Computer-Aided Design 44 (2012) 457–483

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Simultaneous and incremental feature-based multiresolution modeling with feature operations in part design

Sang Hun Lee [a,*], Kunwoo Lee [b,1]

[a] Department of Automotive Engineering, Kookmin University, 861-1, Jeongneung-Dong, Seongbuk-Gu, Seoul, 136-702, Republic of Korea

[b] School of Mechanical and Aerospace Engineering, Seoul National University/I-AMD, San 56-1, Sinlim-dong, Gwanak-gu, Seoul, 151-744, Republic of Korea

ARTICLE INFO

Article history:
Received 22 January 2010
Accepted 22 December 2011

Keywords:
Feature-based design
Multiresolution
Level of detail
Boolean operation
Solid modeling
Non-manifold modeling

ABSTRACT

This paper proposes a new feature-based multiresolution modeling approach that can provide multiresolution representation of dynamically changing CAD models of intermediate design stage. Feature-based multiresolution modeling provides simplified shapes of parts of various levels of detail (LOD) by suppressing the detailed features according to a certain LOD criterion. Unlike previous research having mainly focused on the multiresolution representation of the final design model, our approach can carry out simultaneous and incremental multiresolution representation on the intermediate design models. To implement the system supporting this capability, we developed *history-based selective Boolean operations*, in which if the order of the Boolean operations for a part is altered, the regions affected by the operations are redefined according to the history of the Boolean operations so that the resultant shape may be the same as the original shape of the part. The system implemented using these operations guarantees a unique and valid shape for each intermediate LOD in the simultaneous multiresolution modeling environment. Since the system provides the designer immediately with various detail levels of the CAD model in any design stage, the design process is expected to be accelerated.

1. Introduction

Nowadays, products are more commonly designed by using three-dimensional (3D) computer-aided design (CAD) systems instead of two-dimensional (2D) computer-aided drafting systems [1,2]. However, a detailed 3D CAD model for a product with complicated parts or a large number of parts usually requires vast memory storage [3]. Therefore, it is difficult to manipulate such a detailed model with smooth interaction and rendering, even with advanced computer hardware. To overcome this limitation, the multiresolution modeling technique is highly desirable because it allows object representation at multiple levels of detail.

Two different types of multiresolution modeling techniques have been developed, namely the polygon-based and the feature-based. The polygon-based technique has been studied extensively in computer graphics since the early 1990s, and its main objective is to achieve fast displays [4]. In this technique, an object is represented by a polyhedral model such as a triangular mesh, and the number of faces on the object is reduced when a low resolution model is needed. In contrast, the feature-based technique has been studied just recently in computer-aided design, and its main objective is to provide simplified models for product design and engineering applications [5]. In this technique, an object is represented by a feature-based solid model, and insignificant features of the object are suppressed when a low resolution model is required.

The feature-based multiresolution modeling technique has great potential for application in different areas of engineering, such as product design, analysis, digital mock-up, and virtual manufacturing. In product design, this technique allows the designer to manipulate vast amounts of CAD data more efficiently [6,7]. For instance, a region of the product of interest can be manipulated in full resolution while the remaining portion of the product is displayed in rough resolution. In a network-based collaborative design environment, this technique allows the incremental transmission of a solid model in the unit of design feature [8–10]. This technique enables the system not only to overcome the limitation of the current network bandwidth, but also to share a model at an adequate level of feature detail, depending on the engineering tasks to be undertaken. In engineering analysis, this technique provides a simplified geometric model by suppressing insignificant features [11–14]. Furthermore, in virtual prototyping and manufacturing, it reduces the amount of data storage as it preserves the global shape and simulation accuracy of a part so that real-time rendering and collision detection may be carried out for vast quantities of geometric data [6,7].

* Corresponding author. Tel.: +82 2 910 4835; fax: +82 2 916 0701.
E-mail addresses: shlee@kookmin.ac.kr (S.H. Lee), kunwoo@snu.ac.kr (K. Lee).
1 Tel.: +82 2 880 7141; fax: +82 2 880 8302.

0010-4485/$ – see front matter
doi:10.1016/j.cad.2011.12.005

Research activities in recent years have tried to overcome the challenges of the feature-based multiresolution modeling technique in the product development process [15]. Research has focused on several topics: topological frameworks for representing a multiresolution model, criteria for the level of detail (LOD), generation of valid models after rearrangement of features, the level of the CAD model, and new applications. Regarding topological frameworks, two approaches have been identified: conventional solid boundary representations (*B*-rep) or non-manifold topological (NMT) *B*-reps. The NMT *B*-reps are further classified into the cellular and the mixed-dimensional representation [1,16,17]. For the criteria of the LOD, various types and properties of features have been considered. The criteria most commonly used are the volume of the subtractive feature and the volume of the feature. In the case of the volume of the feature, the feature type may be additive or subtractive. The generation of valid models after feature rearrangement according to the LOD criterion is the most challenging problem. Contemporary approaches for model generation are classified into the delta volume approach [8,18] and the effective-feature-volume approach [5,13] according to the LOD criteria. The effective-feature-volume approach adopts the volume of a feature as the LOD criterion whereas the delta volume approach adopts the volume of the subtractive feature. For the level of the CAD model, part level and assembly level are considered [7]. More extensive and in-depth survey of previous literature is given in the next section.

To provide more efficient design environment with the feature-based multiresolution modeling capabilities, it is necessary to explore the following challenges:

- *Simultaneous multiresolution modeling environment*: the existing feature-based multiresolution modeling approaches have mainly focused on the multiresolution representation of the final design model. In other words, previous researchers assumed that the multiresolution modeling process for a CAD model is generally launched after finishing the product design, and thus, they did not address the multiresolution modeling of dynamically changing CAD models of intermediate design stage. However, if the system allows the designer to immediately see and utilize multiresolution models of various LODs during any stage of the CAD model design, the design task can be carried out more efficiently. To this end, it is necessary to develop a new sophisticated framework and algorithms that can provide a modeling environment in which multiresolution modeling can be performed simultaneously with part modeling, so that multiresolution models may be provided readily at any stage of the design process.
- *Transition feature:* Form features can be classified into volume and transition features according to the underlying geometric modeling operations. Volume features, such as boss, ribe, or hole, are implemented using Boolean operations whereas transition features, such as fillet or taper, are implemented using local modification operations, such as blending or tweaking [1,2]. The existing approaches have mainly focused on volume features [15]. However, most mechanical parts typically have transition features. Therefore, in order for the feature-based multiresolution modeling technique to be acceptable in the design process, it is essential to extend the domain of features to include transition features.
- *Intersection operation*: the existing approaches have only focused on the rearrangement of additive and subtractive features because the basic approach of feature-based modeling has been to create a part model by incrementally adding volumes to or subtracting volumes from the base shape. However, most of the current major CAD systems, such as CATIA [19], Unigraphics [20], and SolidWorks [21], still provide conventional Boolean operations as modeling functions, which include not only union and difference but also intersection. Therefore, in order to apply the feature-based multiresolution modeling methods to the current CAD systems, it is essential to extend these methods to include the intersection operation in addition to the union and difference operations.

To meet the challenges mentioned above, in this paper, we propose the simultaneous and incremental feature-based multiresolution modeling approach based on the *history-based selective Boolean* (*HS-Boolean*) *operations*. In this approach, the feature-based multiresolution modeling task is performed simultaneously with the feature-based part modeling including all types of Boolean operations so that multiresolution models for the current stage of the design model can be provided immediately upon user demand. Since, unlike the normal Boolean operations, the order of the *HS*-Boolean operations is alterable, although union, difference, and intersection are mixed, feature-based multiresolution modeling based on *HS*-Boolean operations can guarantee the same result for arbitrary rearrangement of features and offer a unique and valid shape for each LOD.

The remainder of this paper is organized as follows. Section 2 surveys the related work. Section 3 introduces the simultaneous feature-based multiresolution modeling process and formulates the problem that has to be solved to support this process correctly and efficiently. Section 4 describes the formal definition and properties of the *HS*-Boolean operations. Section 5 introduces the system architecture and data structure for the feature-based design system equipped with multiresolution modeling capabilities. Section 6 describes the algorithms for incremental feature-based multiresolution modeling. Section 7 presents some case studies. Section 8 concludes this paper and suggests future work. The Appendix contains the proofs for important features of this research.

2. Literature survey

Belaziz et al. [22] presented a feature-based tool to aid the integration of design and analysis during the design process. It allows the generation of an abstract analysis model out of a part solid model by morphological analysis of the solid model followed by an abstraction process. In the morphological analysis stage, form features are recognized and a hierarchical feature tree is constructed. In the following abstraction stage, a feature-based model is transformed to an abstract analysis model through a two-phase process: simplification and idealization. The simplification process eliminates any irrelevant features from the feature model to prepare the model for analysis, and the idealization process constructs idealized 3D, 2D, 1D, or 0D geometric entities in accordance with the analysis process used. This paper demonstrates a rearranged hierarchical feature tree and the extraction of simplified models from the tree to illustrate the simplification process. In fact, if the features are rearranged arbitrarily, the resulting shape may not be the same as the original shape, because a change in the order of the union and difference Boolean operations, which the feature modeling operations are converted into, may produce a different resulting shape from the original shape. This is the well-known feature rearrangement problem for feature-based multiresolution modeling [5]. Although Belaziz et al. did not address this problem, they showed the great potential application of the feature-based multiresolution modeling technique in the area of CAD–CAE integration.

Choi et al. [18] firstly studied the feature-based multiresolution representation of a *B*-rep solid model. In their approach, the lowest resolution model is formed by uniting all of the additive features, and then higher resolution models are generated by applying the subtractive features successively in descending order of volume. To implement this process, a hierarchical feature tree for the multiresolution representation is constructed from the original feature history tree. In the hierarchical feature tree, the leaf node at the highest level contains a solid that was created by uniting all of the additive features, whereas the leaf nodes at the lower levels contain solids for the remaining subtractive features. The volume of the subtractive feature interfering with that of the additive features is redefined, because the order of the union and difference operations is not commutative. If a simplified model of a certain LOD is required, the system prunes the branches of the feature tree and performs a boundary evaluation to obtain a corresponding solid model. This method is advantageous because it can be implemented in current commercial 3D CAD systems, which share the same data structure as this method. However, this method has several drawbacks. First, if a part is modeled using only additive features, it can have only a single LOD. Second, this method assumes that subtractive features offer more detail than additive features, although this may not be the case. Third, this method cannot provide adequate LOD models for arbitrary feature rearrangement, regardless of the feature type, either additive or subtractive, because the proposed algorithm supports only the feature arrangement where the additive features take precedence over the subtractive features. Here, an *LOD model* denotes a simplified geometric model at the resolution of a specific LOD and will be used throughout this paper. Finally, this method requires too much computation time to evaluate boundaries because the transition from the current LOD m to the destination LOD n requires $|m-n|$ Boolean operations, which are the most time-consuming operations used in solid modeling.

Koo and Lee [7] proposed a multiresolution modeling method for part and assembly models using the wrap-around operation, which imitates the wrapping of a product with a kitchen plastic wrap. This method is composed of two steps: the first step is the part level wrap-around operation, and the second step is the assembly level wrap-around operation. In the first step, the concave space of each part is found by using the convex inner loop as a clue and is filled by removing the convex inner loop, and then, the faces invisible from outside the part model are removed. The LOD is defined using a set composed of the convex inner loop and the faces that are removed with the convex inner loop. If the wrap-around operation is applied to each part, parts may overlap. In the second step, the faces invisible from outside the assembly model are detected by using the graph traverse method and are deleted. This method is implemented using the Parasolid kernel and it allows arbitrary movement to coarser and finer resolution. An advantage of this method is that assembly models in a commercial CAD system can be simplified according to a given LOD. The limitations of this method are that (1) only the space inside the convex inner loop is deleted even though a concave space exists, (2) the algorithm cannot handle the passages passing through multiple faces, and (3) an assembly model simplified using the wrap-around operations is a mixture of solids and sheets that overlap with each other.

To reduce the computation time required for the extraction of LOD models, Lee et al. [23] introduced the NMT *B*-rep as the topological framework for feature-based multiresolution modeling. In this method, all the features are initially merged into a NMT cellular model, and then, at given LOD, the topological entities constituting the corresponding LOD model are selected and displayed. Since the boundary information of all the features is stored in the NMT cellular model, no geometric calculations are involved in the boundary evaluation. As a result, a solid model at a given LOD can be constructed even more quickly than using the solid-based approaches. They also addressed the problem in which the resulting model after arbitrary feature rearrangement is different from the original, and proposed a modified merge-and-select algorithm based on the NMT *B*-rep as a solution to this problem. From the original merge-and-select algorithm, the select algorithm is modified in consideration of the region affected by each Boolean operation so that the changed order of the Boolean operations yields the same shape as the original shape. However, this approach may produce unacceptable intermediate LOD models. To compensate this drawback, they extended the affecting primitives [24]. However, no proof has been given for the validation of this extension. Furthermore, the algorithm is still highly dependent on the merge-and-select algorithm, and for this reason, this approach is applicable to only NMT *B*-reps except solid data structures.

Lee et al. [8] also developed a feature-based multiresolution modeling method based on the cellular model, and applied it to a network-based collaborative design. They investigated the incremental transmission of solid models through a network and the sharing of the model of adequate LOD for engineering tasks. The ACIS kernel was used to implement the system. However, their study adopted the same LOD criterion and feature rearrangement method as in the approach of Choi et al. [18]. As a result, their method had the same limitations as the approach of Choi et al.: if the features are rearranged in arbitrary order, the method does not guarantee the same resulting shape as the shape of the original solid model. This means that this method is not suitable for various applications. They leave as future work, the extension of the LOD criteria and inclusion of additive features for intermediate LOD models.

Cera et al. [6] developed a new framework for role-based viewing in a collaborative 3D assembly design environment, where multiple users can work simultaneously over the same network. They addressed the access control problem with a combination of multiresolution geometry and access control models. In role-based viewing, each user sees a shared 3D assembly model, in which the constituent components and their sub-features are displayed at various resolutions determined by the users. For role-based viewing, both the polygon-based and the feature-based multiresolution modeling technique are introduced to hide the design detail of a product and to accelerate rendering. First, subtractive features, such as holes, are detected and removed from the original model, and then topology-preserving mesh simplification is applied to the simplified model. However, this method has severe limitations with respect to feature-based multiresolution modeling. First, there is no concept of LOD in the feature suppression step. Second, the features to be removed for multiresolution modeling are limited to only subtractive features, such as holes.

To build LOD models, Kim et al. [25] introduced the feature recognition approach and proposed three operators that can extract a recognized feature. The operators are as follows: wrap-around, smooth-out, and thinning operator. They were implemented based on Parasolid, a commercial solid modeling kernel. By appropriate application of these operators, an assembly model of any desired LOD can be generated. By adopting the feature extraction method, the feature interaction problem can be solved. However, the performance of these multiresolution operators is heavily dependent on their capability to detect multiresolution features. In addition, Kim et al. [25] did not show clearly whether or not the proposed set of multiresolution features is complete and whether or not the extraction operators are robust enough for very complicated solid models.

Lee [5] introduced the concept of an effective volume of a feature to provide valid solids for an arbitrary rearrangement of features, regardless of the feature type. The effective volume of a feature is defined as the actual volume of the feature in the rearranged feature tree when the solid model of the feature is used as a tool body for the Boolean operation. When arranged in the order of feature creation, the effective volume of each feature is defined as the entire volume of the feature. However, the effective volume of a feature can be reduced to a fraction of the original volume after feature rearrangement. Lee described a method to identify the effective volume and provided a mathematical proof of the method's validity. By introducing the concept of an effective volume, we can rearrange features arbitrarily, and select any LOD criteria to suit various applications. However, the effective volume may be defined differently according to the order of relocation of the features. If the order is not selected properly, then some intermediate LOD models may show unacceptable shapes.

To solve this problem, Lee et al. [26] developed a prototype of the history-based selective Boolean operations, in which the volume of each feature is redefined in consideration of the feature type and the current and original orders of the features in order to give a valid and unique model of a specific LOD. However, since these history-based operations support only union and difference excluding intersection, they are unsuitable for implementing feature-based multiresolution modeling for the existing CAD systems that support intersective features based on the intersection operation. To overcome this limitation and widen their application range, a more extensive and in-depth research was further conducted. The final result is presented in this paper.

Lee [13] introduced the mixed dimensional NMT model for multiresolution and multi-abstraction modeling for seamless CAD–CAE integration. Geometric models for analysis are simplified, and idealized models may include not only solid models but also medial surfaces and wireframes. Depending on the engineer's intent and the desired accuracy of the analysis results, geometric models at various levels of abstraction (LOA) need to be provided to the CAE systems. The representation domain from solid to NMT *B*-rep has to be extended to support the multiresolution analysis model in mixed dimensions. In this method, different types of geometric models are simultaneously created for design and analysis for each feature modeling operation, and then they are merged into a master part model, which is a NMT model called a merged set. Next, solid models of various LODs can be immediately extracted from the master model. Moreover, for a specific LOD, abstract NMT models of various LOAs can be rapidly extracted from the master model and transferred to CAE systems. For design changes, the design and analysis models merged into the master model are modified simultaneously and consistently. This CAD/CAE-integrated approach that can provide a unified and simultaneous modeling environment is a good application of the multiresolution NMT modeling technique.

Boundary evaluation is the process that computes the geometric model of the product, i.e. either a solid or a cellular model, from the features that have been specified by the user. Since boundary evaluation has to be executed each time a feature is added, removed, or modified, its efficiency is of paramount importance. There has been a lot of research to improve the efficiency of the boundary evaluation algorithm and its applications [27–29]. For instance, Tilove [28] proposed an efficient method for null-object or same-object detection that requires computationally expensive boundary evaluation. His paper shows that CSG trees representing null objects may be reduced to null trees through the use of a new concept called primitive redundancy, and that tree reduction can be done efficiently by a new technique called spatial localization. In addition, Rossignac and Voelcker [29] proposed the concept of active zones in CSG for accelerating boundary evaluation. They associate with each primitive A in a tree S an active zone Z that represents the region of space where changes to A affect the solid represented by S, and they use a representation of Z instead of S for set-membership classification.

Most modern CAD systems use dual representations: parametric feature-based representation and boundary representation. To take advantages of dual representations, two representations remain consistent at all times. Raghothama and Shapiro [30] tackled the problem of verifying the consistency between them. They presented a way to keep the consistency of the two representations without conversion between them. Their approach is to verify each representation's validity at every step of an update. However, further work still remains as this approach has limitation on the range of modification that may be administered simultaneously to both representations.

3. Problem formulation

3.1. Simultaneous multiresolution modeling with feature modeling

A new design process supporting up-to-date multiresolution models simultaneously with part models realizes the following scenario. First, the user conducts part modeling using the feature-based modeling functions provided by the system. Then, in the background, the system carries out feature-based multiresolution modeling for the current design model. Now, if the user wants to obtain a simplified model, he/she selects an appropriate LOD. Then, the system extracts and displays the LOD model from the multiresolution database. The LOD model can be used for various applications including analysis. If necessary, a series of LOD models for increasing or decreasing order of LODs can be extracted. LOD models can be extracted immediately because the multiresolution database is updated automatically for every feature modeling operation. Fig. 1 shows the feature-based solid modeling process for an example part and a series of LOD models at each feature modeling step. Contemporary feature-based multiresolution modeling methods have never considered such a dynamically changing CAD model. Therefore, in order to support this scenario, it is necessary to develop a new multiresolution modeling method that can minimize the effort for reconstructing multiresolution models and ensure the integrity of the multiresolution models.

3.2. Feature rearrangement problem in simultaneous feature-based multiresolution modeling process

In this section, we define the feature rearrangement problem first, and then apply the previous feature-based multiresolution modeling methods for solving the feature rearrangement problem to the simultaneous multiresolution modeling process. By observing which problems may occur during the feature rearrangement in the process, we can identify the problems that must be solved to achieve simultaneous multiresolution modeling and thus, guarantee the integrity of the multiresolution models.

3.2.1. Feature rearrangement problem in feature-based multiresolution modeling

In feature-based modeling, a feature is a basic modeling unit, and an object is usually modeled by incrementally applying features to the basic shape feature. There are many ways in which features can be classified [2]. In general, form features can be classified into three basic types: volume, transition, and pattern features. For feature-based multiresolution modeling, form features are described by using volumetric representations. The transition and pattern features are converted into volume features. Form features are normally classified into additive and subtractive features, which can be represented by union and difference operations, respectively. However, most of the current major CAD systems, such as CATIA, Unigraphics, and SolidWorks, provide Boolean operations. Therefore, in this paper, we extend the feature types to additive, subtractive, and intersective to support these Boolean operations.

The feature-based modeling process can be represented by a CSG tree whose terminal nodes describe the primitives of the features and the internal nodes represent the regularized Boolean operations. The type of feature determines the Boolean operation applied to it. If a feature is additive, the operation is union ($\cup$); if

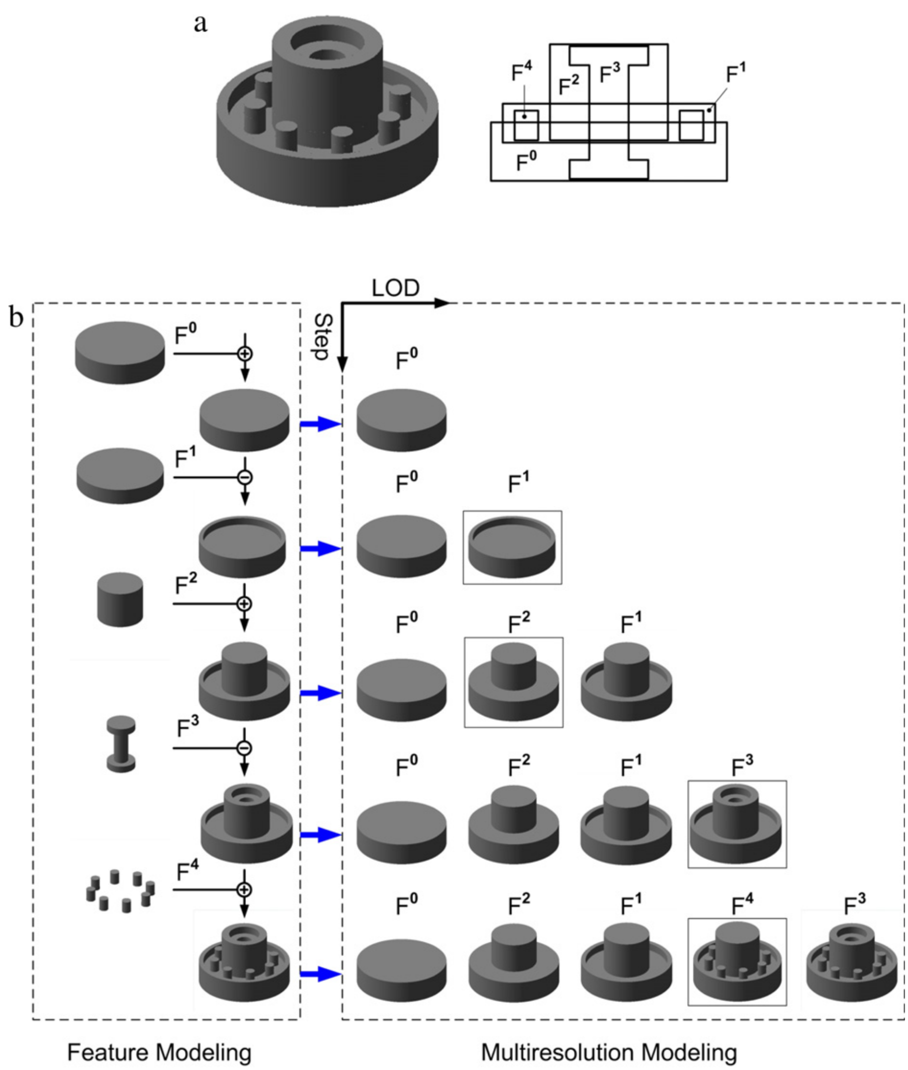

Fig. 1. Simultaneous feature-based multiresolution modeling process in part design: (a) an example part model and its form features, (b) a feature-based modeling process for the part and a set of multiresolution models at each modeling step.

subtractive, the operation is difference ($-$); and if intersective, the operation is intersection ($\cap$). $\cup$ and $\cap$ may be represented by $+$ and $*$ alternatively. The primitive is defined as a point set over the R^3 space in this work, it includes the 3D NMT model as well as the solid model.

Let F, P, and $\otimes$ denote the feature, the primitive of the feature, and the $\cup$, $-$, or $\cap$ Boolean operation of the feature, respectively. If A is one of F, P, or $\otimes$, then A^i denotes the i-th A in the initially created order of the feature, while A_j denotes the j-th A in the currently rearranged order. M_n denotes the resulting model of $n+1$ features. It is created by applying n Boolean operations between the $n+1$ primitives of the features as follows.

$$M_n = \otimes^0 P^0 \otimes^1 P^1 \otimes^2 P^2 \cdots \otimes^i P^i \cdots \otimes^n P^n = \prod_{k=0}^{n} \otimes^k P^k \tag{1}$$

where $\otimes^0 = \cup$, and $\otimes^0 P^0 = \varnothing \cup P^0 = P^0$. As a primitive is normally created first, $\otimes^0 = \cup$ is a common case.

If the features are rearranged, the resulting shape, denoted by M_n', is generally different from the original shape because Boolean operations do not satisfy the associate or commutative law and their result is basically dependent on the operation sequence. M_n' can be represented by the following formula.

$$M_n' = \otimes_0 P_0 \otimes_1 P_1 \otimes_2 P_2 \cdots \otimes_j P_j \cdots \otimes_n P_n = \prod_{\ell=0}^{n} \otimes_\ell P_\ell. \tag{2}$$

For instance, the example part model shown in Fig. 1 is created by applying five features, and the applied Boolean sequence is $P^0 - P^1 \cup P^2 - P^3 \cup P^4$, as illustrated in Fig. 2(a). Let us rearrange the features in the order: $F^0 \to F^2 \to F^1 \to F^4 \to F^3$. The resulting shape of the rearranged Boolean sequence $P^0 \cup P^2 - P^1 \cup P^4 - P^3$ is different from the original, and the intermediate LOD models are not valid as depicted in Fig. 2(b).

Considering the definition of the validity of form features [2], the validity of part models in feature-based multiresolution modeling can be defined as follows:

Definition 1 (*Validity of Part Models in Multiresolution Modeling*)**.** A part model is valid if all the features constituting the part model follow their standard or generic definitions though the features are applied in an arbitrary order to provide multiresolution models. □

To realize feature-based multiresolution modeling, even when the features are rearranged arbitrarily regardless of whether they are additive, subtractive, or intersective, first, the resulting shape must be the same as the original shape, and next, the models of intermediate LODs must have a valid shape. This problem of finding a method satisfying these two conditions is known as the feature rearrangement problem [5], which can be represented shortly as follows:

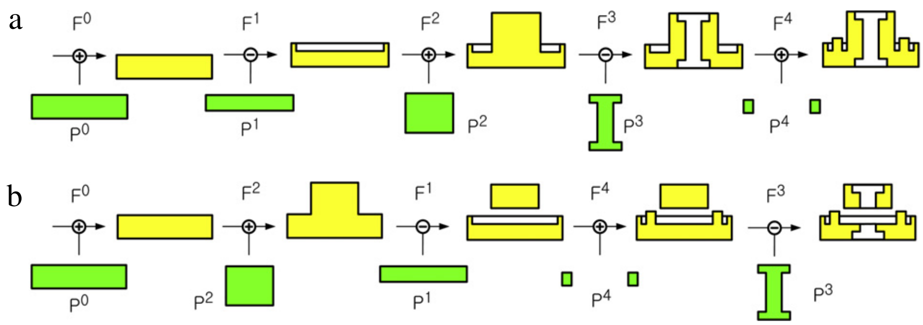

Fig. 2. An example of feature rearrangement: (a) the result of the Boolean operations in the feature creation order, (b) the result of the Boolean operations in the rearranged order.

Definition 2 (*Feature Rearrangement Problem*)**.** The feature rearrangement problem is a problem to find a geometric modeling method satisfying the following conditions:

1. $M'_n = M_n$
2. M'_m, has a valid shape, where $M'_m = \prod_{\ell=0}^{m} \otimes_\ell P_\ell$, $0 < m \le n$

where M_n is the original model which is the result of $n+1$ features in the creation order, and M'_n and M'_m are the LOD models resulting from the $n + 1$ and the $m + 1$ features in the rearranged order, respectively. □

3.2.2. Limitations of previous approaches for feature rearrangement problem in simultaneous feature-based multiresolution modeling environment

In our simultaneous multiresolution modeling process, LOD models at each feature modeling step are defined by rearranging the design features according to an LOD criterion. Although several approaches have been proposed to generate valid models at various LODs after the feature rearrangement, none of them addressed the multiresolution modeling of dynamically changing CAD models of intermediate design stage [15]. Moreover, some approaches even cannot provide adequate LOD models for arbitrary feature rearrangement, regardless of the feature type, either additive or subtractive. For instance, the algorithms proposed by Choi et al. [18] and Lee et al. [8] support only the feature arrangement where the additive features take precedence over the subtractive features and then the subtractive features are rearranged in the descending order of volumes. On the other hand, the other approach based on feature recognition and suppression is not suitable for simultaneous multiresolution modeling. First of all, this approach requires much computation time to generate LOD models, and may extract a quite different set of features for a small change of part shape. Moreover, the existing multiresolution modeling methods of this type has severe limitations. The method proposed by Cera et al. [6] has no concept of LOD in the feature suppression step and remove only subtractive features for simplification of the model. In addition, the approach of Kim et al. [25] does not show clearly whether or not the proposed set of features is complete and whether or not the feature extraction operators are robust enough for complicated solid models. In addition, the performance of the operators is heavily dependent on their capability to detect features.

Currently, the one and only adaptable solution on the feature rearrangement problem, which has been proved theoretically, is the effective-feature-volume approach proposed by Lee [5]. Here, the effective volume of a feature, alternatively, the effective feature volume, represents an adjusted volume of a feature that is obtained by exclusion of some portions of the original volume of the feature. This effective feature volume is used to obtain the same resultant shape as the original shape, regardless of the feature rearrangement. Later, an effective volume of a feature is evolved into an effective zone of a feature, or, alternatively, the effective feature zone, which is represented by a point set over the R^3 space [13]. Therefore, the effective feature zone can represent not only a solid but also a NMT model composed of wireframe, sheet, and solid models. By extending the representation domain from a solid to a NMT modeling space, the feature-based multiresolution modeling technique can be applied to areas of engineering analysis, in which the analysis model is frequently represented by a NMT model. However, even this approach does not take into account dynamically changing CAD models.

In this section, we apply the effective-feature-volume approach to the simultaneous multiresolution modeling process, and observe which problems may occur during the feature rearrangement. The result would let us identify the problems that must be solved to achieve incremental multiresolution modeling and thus, guarantee the integrity of the multiresolution models.

(1) *Different and invalid LOD models depending on feature relocation order.* In the effective-feature-volume approach, if $\otimes$ is one of $\cup$ and $-$, the effective zones of two exchanged features are calculated as follows.

$$P^0 \otimes^1 P^1 \otimes^2 P^2 = P^0 \otimes^2 P^2 \otimes^1 (P^1 - \alpha(1,2)P^2) \tag{3}$$

where

$$\alpha(s,t) = \begin{Bmatrix} 1 & \text{if } \otimes^s \ne \otimes^t \\ 0 & \text{otherwise} \end{Bmatrix}.$$

Here, $\alpha(s,t)P^t = P^t$ if $\alpha(s,t) = 1$, and $\alpha(s,t)P^t = \varnothing$ if $\alpha(s,t) = 0$. As shown in Eq. (3), the effective zone of F^1 is reduced to $P^1 - \alpha(1,2)P^2$ whereas that of F^2 is unchanged. For instance, a model is created by applying three features in the sequence of $F^0 \to F^1 \to F^2$, and the order of features is then changed to $F^0 \to F^2 \to F^1$. If F^0 and F^1 are additive and F^2 is subtractive (i.e., $\otimes^1 = \cup$ and $\otimes^2 = -$), Eq. (3) becomes $P^0 \cup P^1 - P^2 = P^0 - P^2 \cup (P^1 - P^2)$. In this case, the effective zones of F^0, F^1, and F^2 are P^0, $P^1 - P^2$, and P^2, respectively.

The effective zones of all the features in M_n are described by Eq. (4) when a feature at the i-th place, F^i, is relocated to the j-th place.

$$M_n = \left(\prod_{k=0,k\ne i}^{j} \otimes^k P^k\right) \otimes^i \left(P^i - \bigcup_{\ell=1}^{j-i} \alpha(i, i+\ell) P^{i+\ell}\right) \times \left(\prod_{k=j+1}^{n} \otimes^k P^k\right), \quad 0 \le i < j \le n, n \ge 1. \tag{4}$$

For example, a model is created by applying five features in the sequence of $F^0 \to F^1 \to F^2 \to F^3 \to F^4$ whose Boolean sequence is $P^0 - P^1 \cup P^2 - P^3 \cup P^4$, and the order of features is then changed to $F^0 \to F^2 \to F^3 \to F^4 \to F^1$ by moving F^1 to the last location. According to Eq. (4), the Boolean sequence becomes $P^0 \cup P^2 - P^3 \cup P^4 - (P^1 - P^2 - P^4)$. In this case, the effective zones of F^0, F^1, F^2, F^3, and F^4 are P^0, $P^1 - P^2 - P^4$, P^2, P^3, and P^4, respectively.

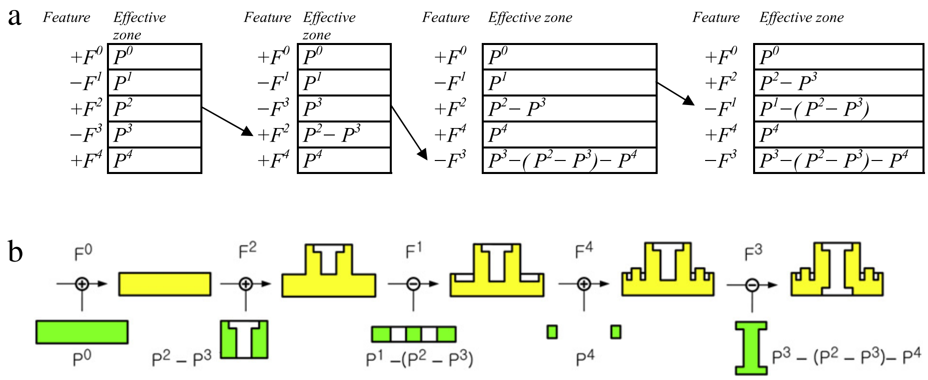

Fig. 3. A feature relocation process and the result of the feature rearrangement, $F^0 \to F^2 \to F^1 \to F^4 \to F^3$: (a) a feature relocation process and the variation of the effective zones of the features for each feature relocation, (b) the LOD models resulting from the feature relocation process.

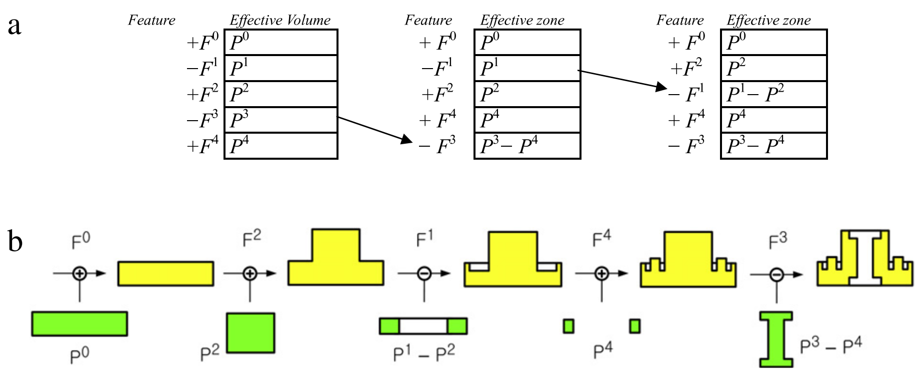

Fig. 4. Another feature relocation process and the result of the feature rearrangement, $F^0 \to F^2 \to F^1 \to F^4 \to F^3$: (a) a feature relocation process and the variation of the effective zones of the features for each feature relocation, (a) the LOD models resulting from the feature relocation process.

For an arbitrary feature rearrangement, the feature relocation operation using Eq. (4) guarantees the same resulting shape as the original shape. On the other hand, in order for the features to be rearranged in a specific order, feature relocation operations may be applied in different orders. Depending on the order of feature relocation operations, the intermediate LOD models may be different, and, in the worst case, some invalid intermediate LOD models may happen.

Figs. 3 and 4 shows the case in which the effective zones of the rearranged features are calculated using Eq. (4) and applied to the rearranged Boolean operations. If the features are moved in the manner shown in Figs. 3(a) and 4(a), then the resulting LOD models are shown in Figs. 3(b) and 4(b), respectively. Although the resulting shapes of the two cases are same, the intermediate LOD models of the two cases are different. The LOD models in Fig. 3 appear invalid, while those in Fig. 4 appear valid. In the case of Fig. 3, the excessive part of the feature effective zone is removed, as on applying F^2, F^1, and F^4, part of the volume, P^3, is subtracted unnecessarily in advance. As shown in Figs. 3 and 4, the intermediate LOD models may alter, depending on the order of the feature relocation operations.

Let us investigate a similar problem in the simultaneous feature-based multiresolution modeling process using the example model shown in Fig. 5. As shown in Fig. 6(b), this model has been created by applying five features in Steps 0–4. In the modeling process, the effective feature zones are redefined by using Eq. (4) as described in Fig. 6(a). Let us assume that the part model is modified by changing the size parameters of the feature F^3 in Step 5 and then removing the feature F^4 in Step 6, as shown in Fig. 6. In each step, the most detailed LOD model is exactly the same as the current CAD model. In Steps 0–4, all of the intermediate LOD models have valid shapes. However, the low resolution models at LOD $= 2$ and 3 in Step 5 and at LOD $= 2$ in Step 6 do not have valid shapes. In Step 5, the feature, F^3, is modified, and thus relocated to the last place. According to Eq. (4), the effective zones of the rearranged features are redefined, as described in Fig. 6(a). However, the new effective zone of the feature, F^2, is invalid because the volume, P^3, is subtracted unnecessarily in advance. As a result, the intermediate LOD models at LOD $= 2$ and 3 have invalid shapes, as shown in Fig. 6(b). In addition, the intermediate LOD models may alter, depending on the order of the feature relocation operations. For example, if the features are relocated one by one in the order of F^3, F^4, F^2, F^1, and F^0, directly from the initial creation order of the features, the effective zones of the features, F^0, F^1, F^2, F^3, and F^4, will be P^0, P^1, P^2, $P^3 - P^4$, and P^4, respectively. This result is different from the one shown in Fig. 6.

The reason for this, is that at each operation, the effective zone of the moved feature is calculated using the current definition of the effective zone. Therefore, to form intermediate LOD models with a valid shape using the operation based on Eq. (4), this order must be selected carefully. Otherwise, it is essential to develop a novel method to calculate the correct effective zones of the features independently of the feature relocation order.

(2) *Inefficiency of feature rearrangement algorithm in simultaneous multiresolution modeling environment.* To prevent any invalid intermediate LOD model from happening in simultaneous multiresolution modeling process, we can adopt the feature rearrangement algorithm proposed by Lee [5]. In the algorithm, feature relocation operations are applied in a strict order as follows: First, the most detailed feature is first selected and moved to the n-th place. Next, the secondly most detailed feature is selected and moved to the $n-1$-th place. This is repeated until the secondly most significant feature is selected and moved to the 1-th place. Consequently, the most significant feature is located at the 0-th place.

In the simultaneous multiresolution modeling process, the effective zones of features need to be redefined with minimum effort

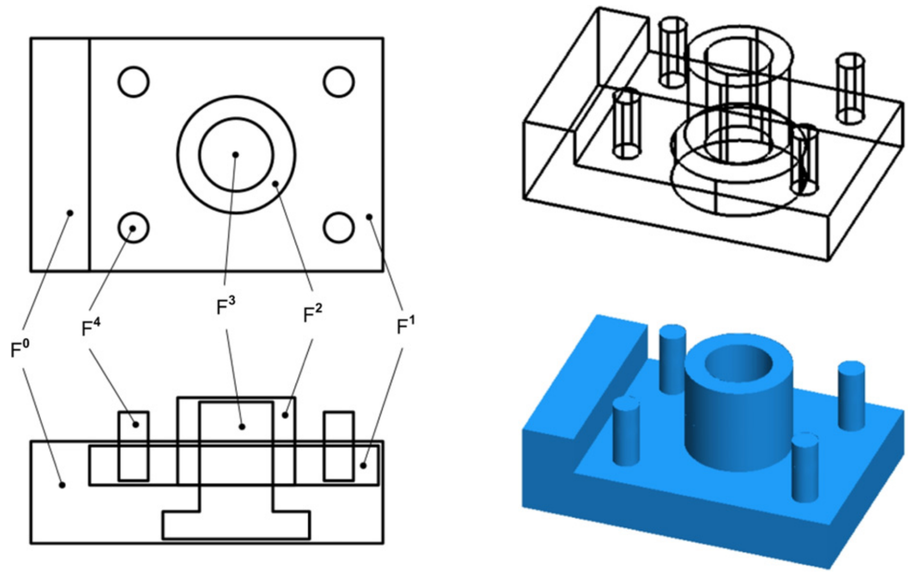

Fig. 5. An example of feature-based solid modeling: the part model and its form features.

at each feature modeling step in order to provide immediately the designer with various LOD models during any stage of the design. However, Lee's algorithm is inefficient because it rearranges all the features one by one in the ascending order of feature significance. In fact, this algorithm has been developed to generate multiresolution models for the final CAD model, not for the dynamically changing CAD models of intermediate design stage. Therefore, it is crucial to devise a method that can redefine the effective feature zones with minimum effort in each feature modeling step, independent of the order of feature relocation operations, and that can guarantee a valid and unique shape for each intermediate LOD model. This is the goal of this work.

4. History-based selective Boolean operations

As a solution to the problems mentioned in the previous section, we proposed the *history-based selective Boolean* (*HS-Boolean*) *operations*, whose order is changeable although the operations are a mixture of union, subtraction, and intersection. Feature operations can be represented by *HS*-Boolean operations according to the feature types. Therefore, for an arbitrary feature rearrangement, the *HS*-Boolean operations will guarantee the same resulting shape as the original shape, and also a unique and valid shape at each intermediate LOD independent of the order of the feature relocations. This section describes the definitions and various properties related with *HS*-Boolean operations.

4.1. NMT models and Boolean operations

In this paper, not only the solid model but also the NMT model is accepted as the geometric model for the feature and the part. In fact, the main applications of feature-based multiresolution modeling are engineering design and analysis. In analysis, solid models are frequently simplified and idealized into lower or mixed dimensional models, which include not only solids but also medial surfaces and wireframes. To support multiresolution models for analysis as well as design, extension of the representation domain from solid to NMT models is essential [13,22]. In addition, under the NMT modeling environment, the feature-based modeling capabilities for both feature deletion and feature interaction detection can be implemented easily [17]. Likewise, the multiresolution modeling capabilities for extracting geometric models at various levels of detail can be carried out in a short time [5,8].

As a mathematical definition for 3D NMT model, in this paper, we accept the cell complex of 3D Euclidean space (E^3) [31], which is a finite collection of an n-cell ($0 \le n \le 3$) such that if e_i and e_j are two different cells in the cell complex, then $e_i \cap e_j = \varnothing$. Mathematically, an n-cell is defined as a bounded subset of E^n that is homeomorphic to an n-dimensional open sphere, and whose boundary consists of a finite number of lower dimensional cells. The 0-cell is equivalent to a vertex, the 1-cell to a edge, the 2-cell to a face, and the 3-cell to a region.

In solid modeling, Boolean operations on solid primitives are expressed and stored in a binary tree, called the CSG (constructive solid geometry) representation of the resulting solid. The leaves of the CSG tree represent instances of simple parameterized primitive solids, and the internal nodes represent potentially more elaborate regular sets and are associated with regularized Boolean operators: union, difference, and intersection, respectively denoted $\cup$, $-$, and $\cap$. A regular set is equal to the topological closure of its interior. The regularized operators perform the standard Boolean operations followed by the topological interior and closure operations, and therefore always return regular sets, which contain their boundary, but no dangling faces, edges, or vertices [29,32].

In NMT modeling, Boolean operations are frequently implemented using the merge-and-select algorithm [31,33]. In this algorithm, all primitives are merged into a single boundary representation, called a *merged set*. The merged set contains a complete description of the input primitives, all intersections between them, and historical information describing the origins of the entities with respect to the topological entities of the original primitives. In the database, all historical information is stored in the cell topological entities. A Boolean logic evaluator, whose input is the CSG representation and the history, selects merged set entities corresponding to the Boolean result. The user can modify Boolean operators or their order of occurrence easily by simply re-executing the selection process. The user can also select with a CSG tree that contains only a subset of the primitives in the merged set, and selectively filter out primitives from the final result without actually removing them from the merged set. If a regular set is required, the algorithm of the purge operation described in Ref. [34] can be applied to the merged set. This algorithm enables quick and arbitrary reshaping of geometric models defined by Boolean operations, and

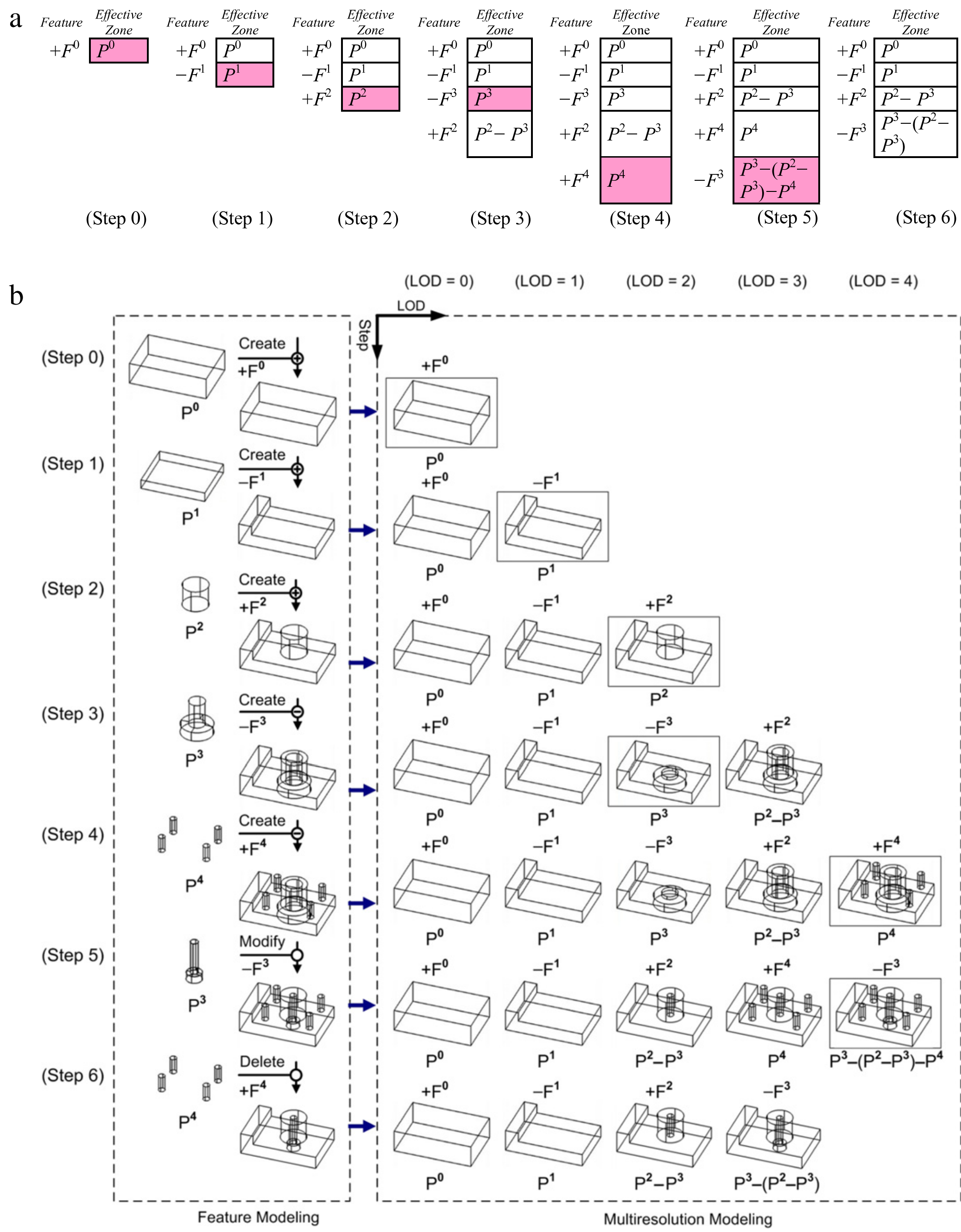

Fig. 6. Application of the effective-feature-volume approach to simultaneous multiresolution modeling for the example shown in Fig. 5: (a) the effective feature zones of the features at each feature modeling step, (b) the feature-based modeling process and the LOD models at each step.

thus can be used for design by trial and error as well as form feature modeling. In this work, all the Boolean operations have been developed based on this algorithm.

4.2. History-based selective Boolean operations

4.2.1. Definition of history-based selective Boolean operations

Boolean operations are basically dependent on the operation sequence, and do not satisfy the associative law or the commutative law. Therefore, to solve the problems formulated in Section 3.2, it is necessary to develop a new type of Boolean operations that can generate the same resulting shape as the original even though the order of the operations is changed. The order independency property enables the features to be rearranged in an arbitrary order while preserving the final shape in the simultaneous feature-based multiresolution modeling process. In addition, this property makes the resultant shape of an LOD model unique and independent of the order of the feature relocation operations. To this end, we propose the *history-based selective Boolean operations* (*HS-Boolean operations*) defined as follows.

Definition 3 (*History-Based Selective Boolean Operations*)**.** Let $\otimes_j^i$, $\hat{\otimes}_j^i$ and P_j^i denote a Boolean operation, a *HS*-Boolean operation, and a primitive, respectively, which is initially applied at the *i*-th place in the order of creation, but currently is applied at the *j*-th place in the rearranged order. If unnecessary, any of *i* and *j* can be omitted.

HS-Boolean operations are the regularized Boolean set operations that satisfy the following condition for M_{j+1}:

$$M_{j-1}\hat{\otimes}_j^x P^x \hat{\otimes}_{j+1}^y P^y = M_{j-1}\hat{\otimes}_j^y P^y \hat{\otimes}_{j+1}^x P^x,$$
$$1 \le j \le n-1, 0 \le x \le n, 0 \le y \le n \tag{5}$$

where n is the total number of the operations, and x and y are arbitrary positions. □

4.2.2. *Effective zone of primitive*

When the order of the Boolean operations is changed, the resulting shape may be different from the original shape because the region affected by each Boolean operation in the initial order of creation is different from the region affected by that in the rearranged order. An *HS*-Boolean operation on its associated primitive can be converted to an Boolean operation and the redefined primitive to provide the same resulting shape.

Observation 1 (*Primitives Likely to Be Redefined for Rearranged Boolean Sequence*)**.** Considering the definition of the validity of feature shape and the properties of Boolean operations, if the rearranged Boolean operation satisfies the following conditions, its primitive is most likely to be redefined:

1. To be located at the post-position in the initial order of creation, but at the pre-position in the rearranged order.
2. To be of a different type—union, difference, or intersection. □

The first condition is necessary to conserve the region affected by each Boolean operation, and the second condition should be considered to prevent excessive exclusion of the volume of the primitive. The redefined primitive is later formulated by Eq. (7), where $\beta(s,t)$ represents the first condition, and $\underline{\otimes}(s,t)$ and $\alpha(s,t)$ describe the second condition.

Definition 4 (*Conversion of HS-Boolean Operations into Boolean Operations*)**.** The effective zone denoted by Z_j^i is the redefined region of the primitive of the Boolean operation, $\otimes_j$, to satisfy the following condition:

$$\prod_{j=0}^{m} \hat{\otimes}_j^{k(j)} P_j = \prod_{j=0}^{m} \otimes_j Z_j^{k(j)}, \quad 0 \le m \le n, 0 \le k(j) \le n, \tag{6}$$

where n is the total number of the operations, and $k(j)$ is the initial location (in the original order) of the current j-th operation (in the rearranged order). □

Proposition 1 (*Effective Zone of Primitive*)**.** *If Z_j^i denotes the effective zone of the primitive of the Boolean operation which is initially applied at the i-th place in the order of creation, but now is applied at the j-th place in the rearranged order, the effective zone is calculated by*

$$Z_j^{k(j)} = \left\{\begin{matrix} P_0, & j=0 \\ P_j \prod_{\ell=0}^{j-1} \underline{\otimes}(j,\ell)\alpha(j,\ell)\beta(k(j),k(\ell))P_\ell, & j>0 \end{matrix}\right\}, \tag{7}$$

where

$$\underline{\otimes}(s,t) = \left\{\begin{matrix} \cap & \text{if } \otimes_s = \cup \text{ and } \otimes_t = \cap \\ \cup & \text{if } \otimes_s = \cap \text{ and } \otimes_t = \cup \\ - & \text{otherwise} \end{matrix}\right\},$$

$$\alpha(s,t) = \left\{\begin{matrix} 1 & \text{if } \otimes_s \ne \otimes_t \text{ and } (\otimes_s = \cup \text{ or } \otimes_t = \cup) \\ 0 & \text{otherwise} \end{matrix}\right\},$$

$$\beta(s,t) = \left\{\begin{matrix} 1 & \text{if } s<t \\ 0 & \text{otherwise} \end{matrix}\right\},$$

and $k(j)$ is the initial location (in the original order) of the current j-th operation (in the rearranged order). □

Table 1
Effective zones of primitives for the exchange of two Boolean operations.

Case	$\otimes^1$	$\otimes^2$	Z^1	Z^2
1	$\cup$	$\cup$	P^1	P^2
2	–	–	P^1	P^2
3	$\cup$	–	$P^1 - P^2$	P^2
4	–	$\cup$	$P^1 - P^2$	P^2
5	$\cap$	$\cap$	P^1	P^2
6	$\cup$	$\cap$	$P^1 \cap P^2$	P^2
7	$\cap$	$\cup$	$P^1 \cup P^2$	P^2
8	–	$\cap$	P^1	P^2
9	$\cap$	–	P^1	P^2

Proof. The effective zone of a primitive for a rearranged Boolean operation, represented by Eq. (7), can be derived from the feature rearrangement algorithm proposed by Lee [5]. In this algorithm, an arbitrary rearrangement of features is decomposed into a sequence of relocations of features, in which the least significant feature is first moved to the last place, followed by the more significant ones in order, until the most significant one is located at the first place. To ensure the same resultant shape as the original, the effective zone of each relocated feature is calculated. For this calculation, in fact, feature relocation is further decomposed into a sequence of feature exchanges, and the effective zones of two features are calculated to guarantee the same Boolean result. Now let us follow this procedure for deriving the effective zones of primitives for an arbitrary rearrangement of Boolean operations.

(*Step* 1) *Exchange of two Boolean operations.* Let $\overset{(k)}{\otimes}{}_j^i$, $\overset{(k)}{P}{}_j^i$, and $\overset{(k)}{Z}{}_j^i$ denote a Boolean operation, a primitive, and an effective zone, respectively, which is initially located at the i-th place in the order of creation, but now is at the j-th place in the k-th moving step. If unnecessary, any of i, j, and k can be omitted. Then, the current result of exchanging two Boolean operations can be represented by Eq. (8). Because there are three different operations, 3×3 cases should be investigated. For these nine cases, the effective zones of primitives are derived by applying the basic laws of Boolean algebra, and the results are summarized in Table 1. The proofs for Cases 1–4 are shown in Ref. [5], and the proofs for Cases 5–9 are described in Appendix A.

$$P^0 \otimes^1 P^1 \otimes^2 P^2 = P^0 \otimes^2 Z^2 \otimes^1 Z^1. \tag{8}$$

Eq. (8) and Table 1 can be represented shortly by Eq. (9).

$$P^0 \otimes^1 P^1 \otimes^2 P^2 = P^0 \otimes^2 P^2 \otimes^1 (P^1 \underline{\otimes}(1,2)\alpha(1,2)P^2) \tag{9}$$

where

$$\underline{\otimes}(s,t) = \left\{\begin{matrix} \cap & \text{if } \otimes_s = \cup \text{ and } \otimes_t = \cap \\ \cup & \text{if } \otimes_s = \cap \text{ and } \otimes_t = \cup \\ - & \text{otherwise} \end{matrix}\right\},$$

$$\alpha(s,t) = \left\{\begin{matrix} 1 & \text{if } \otimes_s \ne \otimes_t \text{ and } (\otimes_s = \cup \text{ or } \otimes_t = \cup) \\ 0 & \text{otherwise} \end{matrix}\right\}.$$

(*Step* 2) *Relocation of a single Boolean operation.* Next, let us investigate the case of relocation of a single Boolean operation. If an arbitrary i-th operation, $\otimes^i P^i$, is moved to the j-th place, the effective zones of all the primitives in M_n are described by Eq. (10). The proof of Eq. (10) appears in Appendix B.

$$M_n = \left(\prod_{k=0,k\ne i}^{j} \otimes^k P^k\right) \otimes^i \left(P^i \prod_{\ell=1}^{j-i} \underline{\otimes}(i,i+\ell)\alpha(i,i+\ell)P^{i+\ell}\right)$$
$$\times \left(\prod_{k=j+1}^{n} \otimes^k P^k\right), \quad i<j. \tag{10}$$

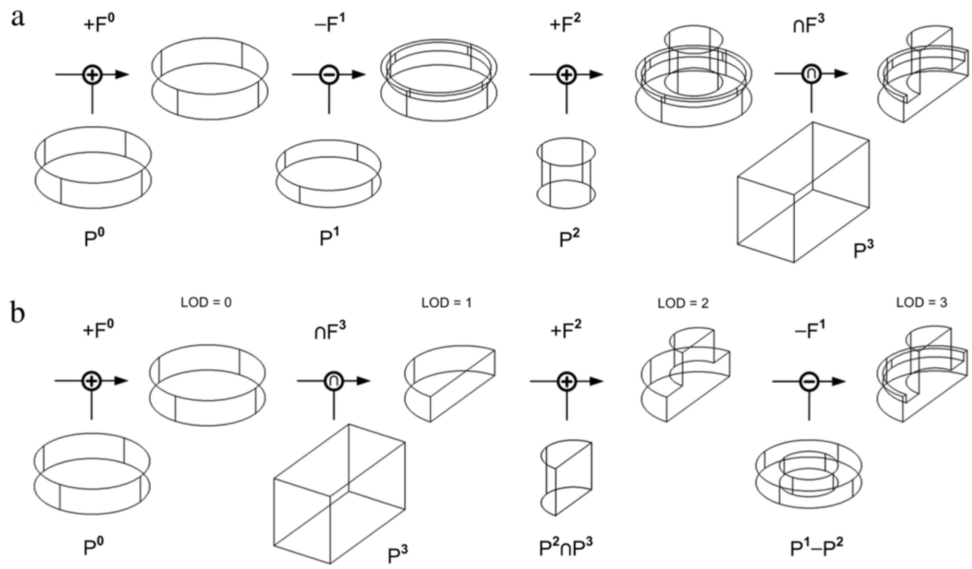

Fig. 7. Rearrangement of the features and the associated Boolean operations using the effective zones of primitives: (a) a feature modeling process, (b) LOD models obtained from the rearranged Boolean operations and the effective zones of primitives.

(*Step* 3) *Rearrangement of all the Boolean operations.* Now, let us rearrange all the Boolean operations accompanied with the primitives according to the feature rearrangement method proposed by Lee [5]. The Boolean operations that are initially arranged in the order of creation are relocated one by one in a specified order, starting with the one determined to be located last. This relocation operation is repeated until all the operations are located at their positions. Whenever each operation and primitive is moved to its new place, its new effective zone of the primitive is redefined using Eq. (10). If the Boolean operations are rearranged following this algorithm, the effective primitive zones for all the Boolean operations can be obtained, as described by Eq. (11). The proof of Eq. (11) appears in Appendix C.

$$M_n = \otimes_0 P_0 \prod_{j=1}^{n} \otimes_j \left(P_j \prod_{\ell=0}^{j-1} \underline{\otimes}(j,\ell)\alpha(j,\ell)\beta(k(j),k(\ell))P_\ell \right) \tag{11}$$

where $\beta(s,t) = \begin{Bmatrix} 1 & \text{if } s<t \\ 0 & \text{otherwise} \end{Bmatrix}$, and $k(j)$ is the initial location (in the original order) of the current j-th feature (in the rearranged order). □

Case study for effective zone of primitive. Let us apply Eqs. (6) and (7) to the example part shown in Fig. 1(a). Let us suppose that the first three features are applied, and then an intersective feature is applied to get the cross-sectional view of the part, as shown in Fig. 7(a). If the volume of a feature is selected as an LOD criterion, the order of the features is altered to $F^0 \to F^3 \to F^2 \to F^1$. The rearranged feature operations can be represented by the HS-Boolean operations and their primitives, $\hat{+}_0^0 P^0$, $\hat{*}_1^3 P^3$, $\hat{+}_2^2 P^2$, and $\hat{-}_3^1 P^1$. These HS-Boolean operations and the primitives are converted to the Boolean operations and the effective zones of the primitives, $+P^0$, $*P^3$, $+(P^2 * P^3)$, and $-(P^1 - P^2)$, respectively, according to Eqs. (6) and (7). As a result, a sequence of LOD models are generated as shown in Fig. 7(b).

Observation 2 (*Effective Zone of Primitive for Union or Difference Operation*)**.** If only union and difference operations are used in part modeling, Eq. (7) for union, difference, and intersection can be reduced to Eq. (12).

$$Z_j^i = \begin{Bmatrix} P_0^i, & j=0 \\ P_j^i - \bigcup_{\ell=1}^{j-1} \alpha(j,\ell)\beta(i,k(\ell))P_\ell, & j>0 \end{Bmatrix}, \tag{12}$$

where $\alpha(s,t) = \begin{Bmatrix} 1 & \text{if } \otimes_s \neq \otimes_t \\ 0 & \text{otherwise} \end{Bmatrix}$, $\beta(s,t) = \begin{Bmatrix} 1 & \text{if } s<t \\ 0 & \text{otherwise} \end{Bmatrix}$, and $k(j)$ is the initial location (in the original order) of the current j-th feature (in the rearranged order). □

If HS-Boolean operations are applied to the case of Fig. 6, the effective zones of the features become as shown in Fig. 8(a). In particular, in Step 5, since the volume of F^2 is preserved, valid LOD models can be obtained at LOD = 2 and 3, as illustrated in Fig. 8(b).

4.2.3. Exchangeability of two adjacent HS-Boolean operations

According to Definition 4, HS-Boolean operations on their primitives are represented by the same signs of Boolean operations on the effective zones of their primitives calculated by Eq. (7). In this section, we will show that this converted representation satisfies the definition of HS-Boolean operations shown in Definition 3.

Proof of Eq. (5). According to the algorithm described in Appendix C, the model M_{j+1} at LOD $= j+1$ can be represented by $M_{j+1} = \prod_{i=0}^{j+1} \overset{(n-j-1)}{\otimes_i} \overset{(n-j-1)}{P_i}$. Let us assume that in Step $n-j-1$, the locations of P^x and P^y are arbitrary s ($0 \le s \le j+1$) and t ($0 \le t \le j+1$), respectively. If P^y and P^x are moved to the $j+1$-th and j-th place in turn, then M_{j+1} in Step $n-j+1$ can be represented as follows

$$\begin{aligned} M_{j+1} &= \prod_{i=0}^{j+1} \overset{(n-j-1)}{\otimes_i} \overset{(n-j-1)}{P_i} \\ &= \left(\prod_{i=0, i\neq s, i\neq t}^{j+1} \overset{(n-j-1)}{\otimes_i} \overset{(n-j-1)}{P_i} \right) \otimes^x Z_j^x \otimes^y Z_{j+1}^y \\ &= \left(\prod_{i=0}^{j-1} \overset{(n-j+1)}{\otimes_i} \overset{(n-j+1)}{P_i} \right) \otimes^x Z_j^x \otimes^y Z_{j+1}^y \\ &= \left(\prod_{i=0}^{j-1} \overset{(n-j+1)}{\otimes_i} \overset{(n-j+1)}{P_i} \right) \hat{\otimes}_j^x P^x \, \hat{\otimes}_{j+1}^y P^y. \end{aligned} \tag{13}$$

Since Eq. (13) is equal to the left side of Eq. (5), M_{j-1} can be represented by Eq. (14).

$$M_{j-1} = \prod_{i=0}^{j-1} \overset{(n-j+1)}{\otimes_i} \overset{(n-j+1)}{P_i}. \tag{14}$$

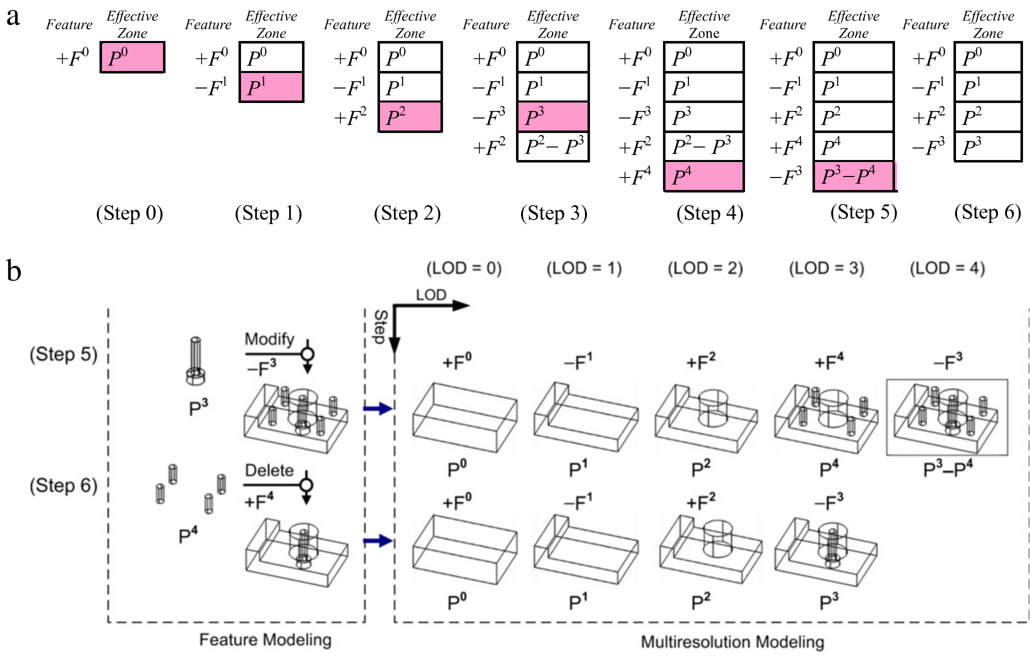

Fig. 8. Relocation of feature F^3 to the last place due to feature modification: (a) the effective zones of the features in each step, (b) the shapes of LOD models.

Next, let us apply the relocation operation in a different order. If P^x and P^y are moved to the $j+1$-th and j-th places respectively and Eq. (14) is applied, then M_{j+1} can be represented as follows.

$$\begin{aligned}
M_{j+1} &= \prod_{i=0}^{j+1} {}^{(n-j-1)}\otimes_i \, {}^{(n-j-1)}P_i \\
&= \left(\prod_{i=0, i\neq s, i\neq t}^{j+1} {}^{(n-j-1)}\otimes_i \, {}^{(n-j-1)}P_i \right) \otimes^y Z_j^y \otimes^x Z_{j+1}^x \\
&= \left(\prod_{i=0}^{j-1} {}^{(n-j+1)}\otimes_i \, {}^{(n-j+1)}P_i \right) \otimes^y Z_j^y \otimes^x Z_{j+1}^x \\
&= \left(\prod_{i=0}^{j-1} {}^{(n-j+1)}\otimes_i \, {}^{(n-j+1)}P_i \right) \hat{\otimes}_j^y P^y \hat{\otimes}_{j+1}^x P^x \\
&= M_{j-1} \hat{\otimes}_j^y P^y \hat{\otimes}_{j+1}^x P^x. \qquad (15)
\end{aligned}$$

Eq. (15) is equal to the right side of Eq. (5). From Eqs. (13) and (15), the following relationship, which is shown in Eq. (5), is established, regardless of the operation type.

$$M_{j+1} = M_{j-1} \hat{\otimes}_j^x P^x \hat{\otimes}_{j+1}^y P^y = M_{j-1} \hat{\otimes}_j^y P^y \hat{\otimes}_{j+1}^x P^x. \quad \square$$

The exchangeable property of the *HS*-Boolean operations is very useful for solving the feature rearrangement problem. Once feature modeling operations are converted to *HS*-Boolean operations, the *HS*-Boolean operations can be rearranged in an arbitrary order while preserving the final shape. Therefore, a feature-based multiresolution modeling method based on *HS*-Boolean operations always satisfies the first condition of the feature rearrangement problem described in Definition 2, $M_n' = M_n$.

4.2.4. Uniqueness of the result of *HS*-Boolean operations

In our approach for feature-based multiresolution modeling, a sequence of the rearranged features are converted to their corresponding *HS*-Boolean operations and their primitives. For a given LOD, a multiresolution model is defined by the subset of the *HS*-Boolean operations in the rearranged order. Let us return these *HS*-Boolean operations in the initial order of creation by exchanging two adjacent operations repeatedly. In virtue of the exchangeability property of *HS*-Boolean operations, as shown in Eq. (5), the resulting shape is not changed after this rearrangement. Then, from Eq. (7), the effective zone of the j-th operation becomes $Z_j^i = P^i$, because $\beta(i,j) = 0$ as $i = j$. The multiresolution model at LOD $= m$ is represented by Eq. (16), which is derived from Eq. (6) using $Z_j^i = P^i$.

$$M_m = \prod_{j=0}^{m} \hat{\otimes}_j^{k(j)} P^{k(j)} = \prod_{j=0}^{m} \otimes^{k(j)} P^{k(j)}, \quad 0 \le m \le n. \qquad (16)$$

This equation means that the result of the *HS*-Boolean operations on the refined feature volumes in the rearranged order is always the same as that of the Boolean operations on the full feature volumes in the original order. Consequently, the resultant shape of an LOD model is unique and independent from the order of the feature relocation operations. Therefore, a multiresolution modeling method based on *HS*-Boolean operations always satisfies the second condition of the feature rearrangement problem described in Definition 2 in Section 3.2.1.

Let us investigate this property using the example model shown in Fig. 7(a). As shown in Fig. 7(b), the features are rearranged into $F^0 \rightarrow F^3 \rightarrow F^2 \rightarrow F^1$ according to an LOD criterion. As illustrated in Table 2, each LOD model is defined by a sequence of *HS*-Boolean operations on the feature primitives in the rearranged order. This sequence can be converted to equivalent Boolean operations on the effective zones of the primitives in the original order of creation according to Eq. (16). For example, the multiresolution model at LOD $= 2$ is defined as $\hat{+}_0^0 P^0 \hat{*}_1^3 P^3 \hat{+}_2^2 P^2 = \hat{+}_0^0 P^0 \hat{+}_1^2 P^2 \hat{*}_2^3 P^3 = P^0 + P^2 * P^3$. Table 2 and Fig. 9 describe the *HS*-Boolean operations and their equivalent Boolean operations to define each LOD model.

Using the property shown in Eq. (16), it is possible to obtain an LOD model by suppressing the irrelevant features from the feature tree without changing the order of the features. This approach allows easy implementation of feature-based multiresolution modeling in current commercial CAD systems because transition features do not have to be converted to volume features. However, this approach requires more computation time for the transition of the current LOD model to the next LOD model than the approach based on *HS*-Boolean operations, in which LOD models are generated incrementally.

Table 2
HS-Boolean operations on the feature primitives in the rearranged order and in the initial order of creation, and their corresponding Boolean operations on the effective zones of the primitives, to define the LOD models of the example model.

LOD	Feature	*HS*-Boolean op.	Effective zone	Definition of LOD model			
				HS-Boolean ops. on the feature primitives in the rearranged order		*HS*-Boolean ops. on the effective zones in the order of creation	
				HS-Boolean ops. on primitives	Boolean ops. on effective zones	*HS*-Boolean ops. on primitives	Boolean ops. on effective zones
0	$+F^0$	$\hat{+}^0_0P^0$	P^0	$\hat{+}^0_0P^0$	P^0	$\hat{+}^0_0P^0$	P^0
1	$*F^3$	$\hat{*}^3_1P^3$	P^3	$\hat{+}^0_0P^0\hat{*}^3_1P^3$	$P^0 * P^3$	$\hat{+}^0_0P^0\hat{*}^3_1P^3$	$P^0 * P^3$
2	$+F^2$	$\hat{+}^2_2P^2$	$P^2 * P^3$	$\hat{+}^0_0P^0\hat{*}^3_1P^3\hat{+}^2_2P^2$	$P^0 * P^3 + (P^2 * P^3)$	$\hat{+}^0_0P^0\hat{+}^2_1P^2\hat{*}^3_2P^3$	$P^0 + P^2 * P^3$
3	$-F^1$	$\hat{-}^1_3P^1$	$P^1 - P^2$	$\hat{+}^0_0P^0\hat{*}^3_1P^3\hat{+}^2_2P^2\hat{-}^1_3P^1$	$P^0 * P^3 + (P^2 * P^3) - (P^1 - P^2)$	$\hat{+}^0_0P^0\hat{-}^1_1P^1\hat{+}^2_2P^2\hat{*}^3_3P^3$	$P^0 - P^1 + P^2 * P^3$

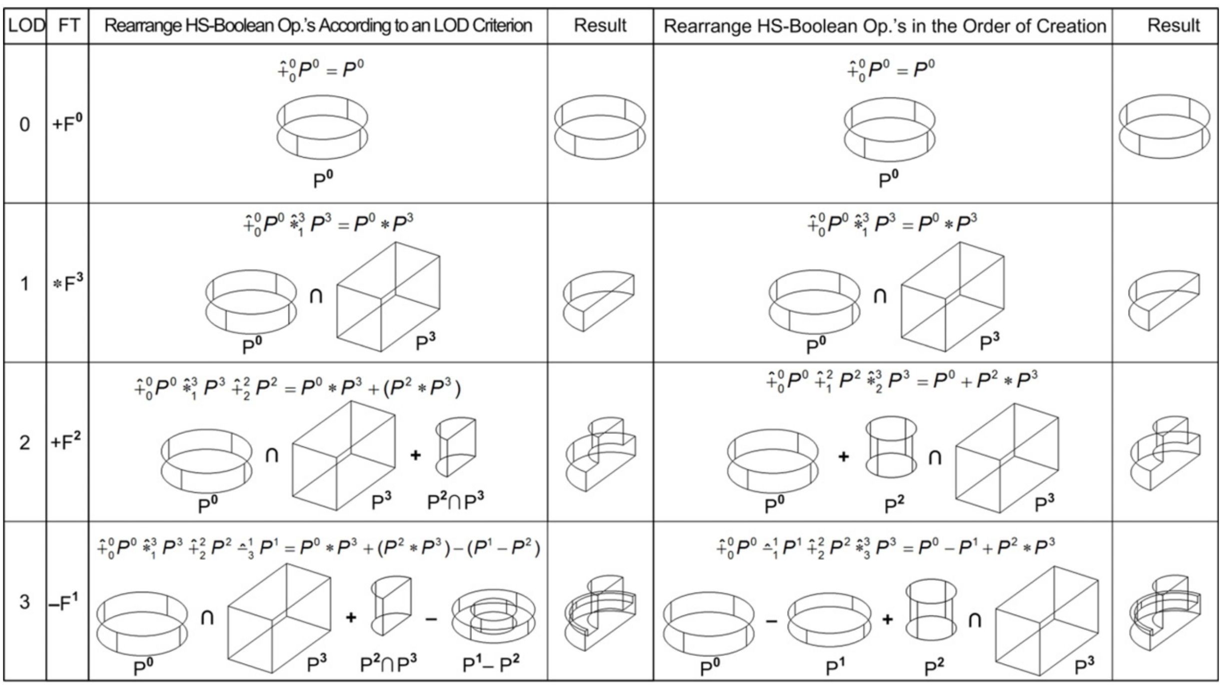

Fig. 9. *HS*-Boolean operations on the primitives in the rearranged order and in the initial creation order, and their corresponding Boolean operations on the effective zones of the primitives, to generate the LOD models of the example model.

5. System architecture and data structure

5.1. System interface and architecture

To support the simultaneous multiresolution modeling process or scenario proposed in Section 3.1, we implemented a feature-based NMT modeling system based on the *HS*-Boolean operations as shown in Fig. 10. In the context of simultaneous multiresolution modeling, the system has to update the multiresolution representation in each feature modeling step in order to provide the LOD models for the current CAD model immediately upon user demand. The system has two levels of command interfaces: the user level and the system level.

- *User-level commands:* they are given directly by the user and consist of the feature-based modeling and the multiresolution modeling commands. The feature-based modeling commands include a variety of commands for creation, modification, and deletion of features. The feature-based multiresolution modeling commands consist of various commands for changing the LOD criterion, assigning feature priority, updating multiresolution representation, displaying LOD models, and so on.
- *System-level commands:* they are executed internally by the system include database management commands for form features, multiresolution features, and NMT models. The database and its management modules are described in more detail in the following paragraph. A single user-level command can be converted into one or more system-level commands. When the user gives a user-level feature-based modeling command, the system executes not only a feature modeling function but also a multiresolution modeling function like 'update the multiresolution model'.

To accommodate the simultaneous multiresolution modeling paradigm and to execute the commands mentioned above, we implemented a feature-based NMT modeling system, which is a revised version of the system developed in Ref. [5]. Like the previous system, to facilitate the implementation, our proposed system stores and manipulates three types of data: a NMT model, which contains all geometric data for the multiresolution model; a graph of the form features for feature-based solid modeling; and a list of multiresolution features whose attributes contain all the necessary information to build a multiresolution solid model and extract LOD models from it. A multiresolution feature is called *MR-feature* for short in this paper. The attributes of an *MR*-feature

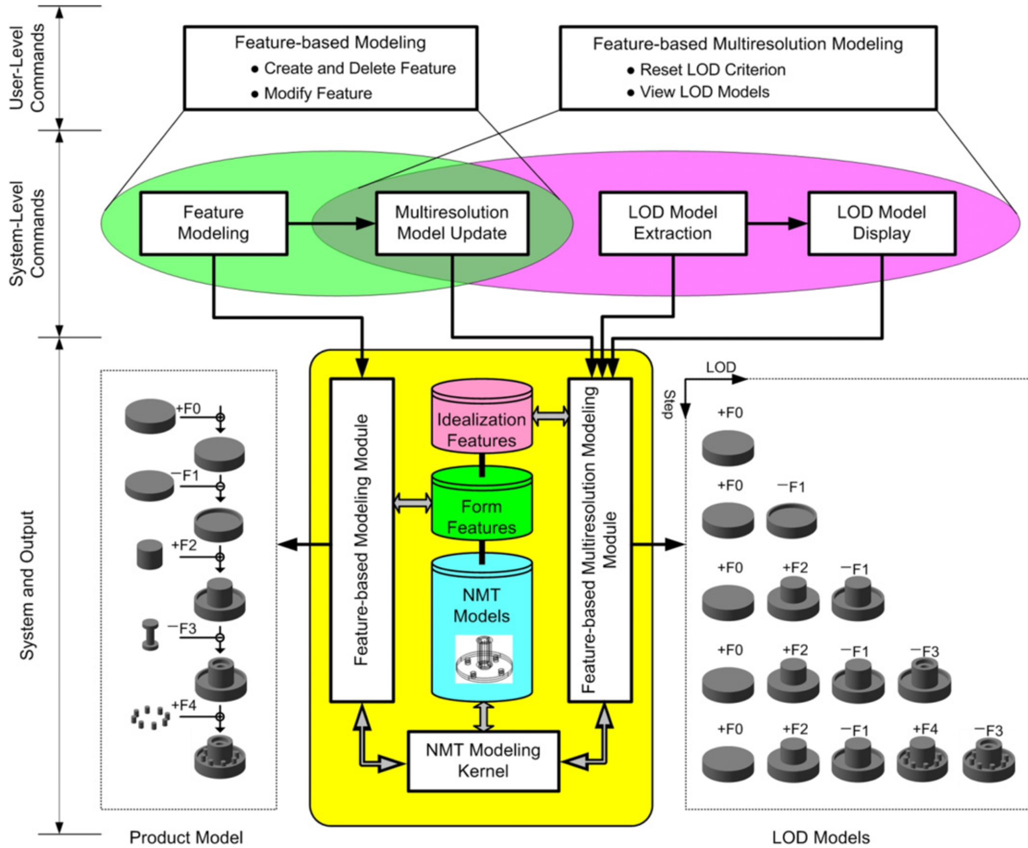

Fig. 10. Proposed system architecture and command interface to support simultaneous feature-based multiresolution modeling with feature modeling in part design.

include the LOD, the pointer to the form feature, the creation order, the type of Boolean operation, the name of the feature primitive, the effective zone of the feature, the definition of the LOD model, and so on. *MR*-features are arranged in the order of LODs. As shown in Fig. 10, these three types of database are manipulated by three main modules as follows:

- *NMT modeling kernel*: it creates and manipulates all of the geometric models for design and for other downstream applications. This kernel was developed based on the Partial Entity Structure [16]. To facilitate the implementation of feature-based multiresolution modeling, a merged set [31,33] of the feature primitives is also introduced, which contains all geometric data to be used for the creation of the multiresolution model.
- *Feature-based modeling module*: it manages the library and database of the form features during the life cycles of the features. In each feature modeling operation, the solid model of the design feature is merged into the master part model, which is a NMT model. Currently, a limited number of features, such as boss, rib, or hole, are implemented in the system. The feature library will be extended in the future.
- *Feature-based multiresolution modeling module*: it manages the *MR*-features and performs the detail removal tasks required to obtain simplified LOD models. If the user specifies an LOD, then the corresponding solid model is extracted from the master model. It is also possible to extract a series of solid models at higher or lower LODs. These multiresolution models can be used for various applications. This module is re-implemented using the *HS*-Boolean operations proposed in this paper.

5.2. Data structure

The part model contains the feature and geometry information. Fig. 11 shows the schematic diagram of the part model described in the EXPRESS-G style. This diagram represents only the entities and attributes referred to in this paper. The part model also records the feature modeling history in its attributes. The following section explains the features, together with the merged set introduced in the system, in more detail.

5.2.1. A merged set model in non-manifold topology

An NMT model is adopted as the topological framework for part representation, and the *merge-and-select* algorithm is introduced for boundary evaluation. The system has been developed based on the NGM (Non-manifold Geometric Modeller), which is a NMT modeling kernel based on the Partial Entity Structure [16] developed by the authors. The part master model has a merged set of all of the primitives of the features. In order to support the *merge & select* algorithm, the merged set should contain a complete description of the input primitives and all intersections between them, along with historical information describing origins of the entities in terms of the topological entities of the original primitives. To meet this requirement, the proposed data structure, as shown in the schematic diagram in Fig. 11, stores the historical information in the Ownership attribute of the Cell Entity class. The origin of an entity in a merged set can be classified into two categories: *inheritance* from a partial or whole entity of a feature primitive, or *intersection* between the entities of feature primitives. The Ownership class has two attributes, denoted *part_of* and *intersection_of*, for these two cases. The *part_of* attribute stores cell entities at the lowest dimensionality in the corresponding

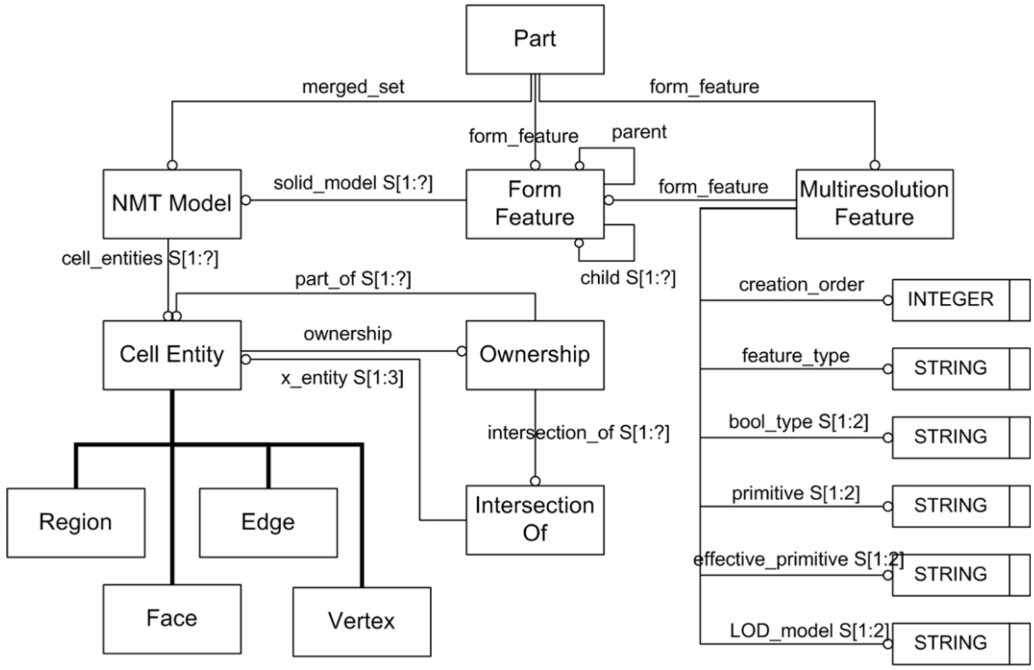

Fig. 11. Data structure for parts.

primitives, and the number of entities is the same as the number of primitives participating in the Boolean operations. The *intersection_of* attribute stores the cell entities participating in the intersection. The entities recorded in the *intersection_of* attribute are also stored redundantly in the *part_of* attribute to facilitate implementation of the *merge & select* algorithm.

Using the properties of the merged set, the feature-based modeling capabilities for both feature deletion and feature interaction detection can be implemented easily [17,35]. Likewise, the multiresolution modeling capabilities for extracting geometric models at various levels of detail can be carried out in a short time.

5.2.2. *Form features*

Feature taxonomies can be based on either product categories, intended applications of the features, or feature shapes. In present, there are no universally accepted or widely used feature taxonomies. Several taxonomy schemes have been proposed for classification entirely by feature shape [2,36]. In general, form features can be classified into three basic types: volume, transition, and pattern features. A volume feature is an increment or decrement to the volume of a shape, such as a hole or a boss. A transition feature separates or blends surfaces, such as fillets, drafts, or offsets. A feature pattern is a set of similar features in a regular geometric arrangement, such as a circular or array pattern [19–21,37].

For feature-based multiresolution modeling, form features are described using volumetric representations, and classified into additive and subtractive features. The transition and pattern features are converted into volume features and reclassified into additive and subtractive features in the following manner.

- A feature pattern is converted into a volume feature representing the resulting shape of the pattern. If the objective feature to which pattern is applied is additive, the volume feature is additive. Otherwise, the volume feature is subtractive.
- A transition feature is also converted into a volume feature representing the volume subtracted or added to a part shape, and classified as a subtractive or additive feature accordingly. However, for a tweaking feature that replaces the geometry of a face with a new surface, the transition feature can be converted into two different types of volume features: a subtractive and an additive one. Let M_i $(0 \le i \le n)$ denote the model at the i-th step. If ΔM_i^+ and ΔM_i^- defined respectively as:

$$\Delta M_i^+ = M_i - M_{i-1} \tag{17}$$

$$\Delta M_i^- = M_{i-1} - M_i \tag{18}$$

then M_i can be defined by

$$M_i = M_{i-1} \cup \Delta M_i^+ - \Delta M_i^- \tag{19}$$

ΔM_i^+ and ΔM_i^- are stored for transition features in the *MR*-feature database.

5.2.3. *Multiresolution features*

In order to facilitate the implementation of feature-based multiresolution modeling, the *MR*-feature proposed in Ref. [5] is introduced and extended to support transition features. The *MR*-feature table contains all necessary information for the construction of a multiresolution model and for the extraction of LOD models from it. The attributes of the *MR*-feature include the pointer to the form feature, the creation order, the feature type, the type of the Boolean operation, the name of the feature primitive, the effective zone of the feature, the definition of the LOD model, and so on. Note that the *MR*-feature is extended to store two Boolean operations and their associated data in order to support transition features.

For instance, when the example solid model is created as shown in Fig. 12, the *MR*-feature table is filled like those shown in Table 3. The model contains two volume features, F^0 (Base) and F^1 (Step), and three transition features, F^2 (Tweak), F^3 (Fillet), and F^4 (Round). F^2 is converted to a pair of additive and subtractive volume features whereas F^3 and F^4 are converted to an additive and a subtractive volume feature, respectively. In the *MR*-feature table, the *MR*-features are initially arranged in the order of feature creation. The feature type is currently one of three cases: volume,

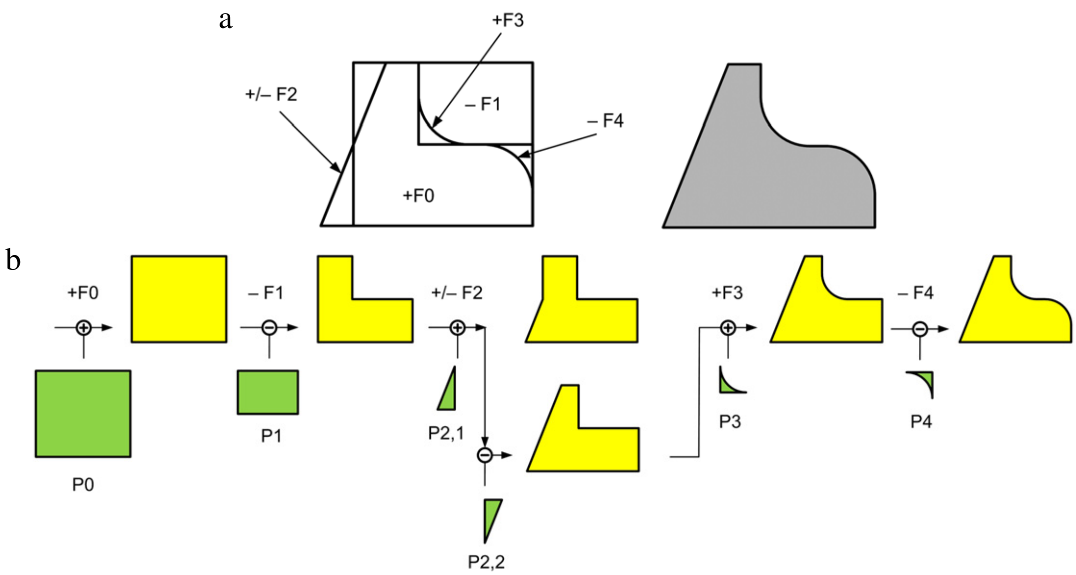

Fig. 12. An example of part modeling with volume and transition features: (a) a part model and its form features, (b) feature-based modeling process to create the part model.

Table 3
Initial *MR*-feature table for the example part model shown in Fig. 12.

No	Feature name	Creation order	Feature type	Bool [2]	Primitive [2]	Effective zone [2]	LOD model [2]
0	Base	0	V	$+$	$P0$	$P0$	$P0$
1	Step	1	V	$-$	$P1$	$P1$	$P0 - P1$
2	Tweak	2	T	$+$	$P2,0$	$P2,0$	$P0 - P1 + P2,0$
				$-$	$P2,1$	$P2,1$	$P0 - P1 + P2,0 - P2,1$
3	Fillet	3	T	$+$	$P3$	$P3$	$P0 - P1 + P2,0 - P2,1 + P3$
4	Round	4	T	$-$	$P4$	$P4$	$P0 - P1 + P2,0 - P2,1 + P3 - P4$

transition, or pattern, and are denoted by V, T, or P, respectively. The effective zone of each feature is assigned to the name of its geometric model. This means that, in the case of the current feature arrangement, the whole of the feature geometry is used by the Boolean operations to extract LOD models. In the multiresolution modeling processes, the *MR*-features are arranged in the order of LODs, and their effective zones are redefined following Eq. (7). In the case of transition features, they can have two geometric models for union and subtraction in maximum. As illustrated in Fig. 12, the tweak feature F_2 stores two set of data for union and difference operations. The LOD model is defined by a sequence of Boolean operations on the effective zones, and finally extracted from the NMT model using the merge-and-select algorithm.

6. Algorithms for multiresolution feature modeling

6.1. LOD criteria

The LOD criteria for features can be classified into two types: the absolute and the relative criterion.

First, the absolute criterion is a measure that is independent from the previously selected features, and does not require any reference model with which the candidate features are compared. This criterion usually originates from the feature property itself, like the volume of a feature. When a feature is created or modified, the new location of its corresponding *MR*-feature is found first, and then a range of *MR*-features is updated accordingly. If we choose an absolute LOD criterion, the range of the features to be relocated is determined straightforwardly.

Next, a relative criterion is a measure that is influenced by the features selected previously in the feature arrangement algorithm. It requires a reference model, which is usually the model of the final or initial part shape or the LOD model at one higher or lower LOD. A typical relative criterion is the volume difference between a reference model and a candidate model. The candidate models are generated by including or excluding candidate features to the LOD model determined in the previous step. The candidate model showing minimal difference with the reference model is found and its corresponding feature is selected as the feature at the current LOD. In this case, the range of the *MR*-features affected by the relocation of the modified or created *MR*-feature is not determined straightforwardly. A forced but safe way is to adopt the whole range as the range of the *MR*-features to be redefined. If a NMT framework is introduced for multiresolution representation, feature rearrangement can be carried out very efficiently even if a relative criterion is adopted, because no boundary evaluation would be required to generate a lot of candidate LOD models.

In the previous effective-feature-volume approach [5], though features can be rearranged arbitrarily regardless of the feature type, the order of relocation of the features should be selected carefully to prevent unacceptable shapes of intermediate LOD models. As a result, the model of the initial part shape or the LOD model at one lower LOD cannot be a reference model. However, in our approach, any kind of model including the model of the initial or final part shape and the LOD model at one higher or lower LOD is available as a reference model, and the shape of each LOD model is determined uniquely and independently of the order of the feature relocations.

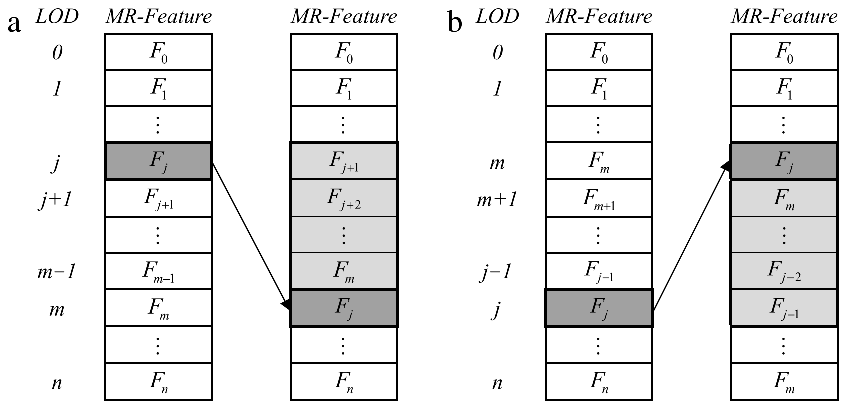

Fig. 13. Modification of an *MR*-feature: (a) the range of affected *MR*-features when the feature is moved backward, (b) the range of affected *MR*-features when the feature is moved forward.

6.2. Creation, modification, and deletion of multiresolution features

In this section, we describe the algorithm for feature rearrangement for an absolute LOD criterion. This algorithm is the same as that for a relative LOD criterion except that the range of the updated *MR*-features is set to the whole range. Feature-based modeling operations are classified into three types: creation, modification, and deletion of a feature. In order to support multiresolution modeling simultaneously, the current multiresolution representation should be updated immediately whenever each feature operation is carried out. Feature creation and deletion algorithms are exactly the same as the feature modification algorithm, except for one additional step that creates or deletes an *MR*-feature.

6.2.1. Feature modification

If a feature is modified, the corresponding *MR*-feature needs to be relocated to a different place according to the LOD criterion. As shown in Fig. 13, if the j-th *MR*-feature, F_j, is relocated to the m-th place due to feature modification, the redefined *MR*-features that would need to be redefined would range from j to m if $j < m$, or from m to j if $j > m$.

Let us assume that the operations to manage the records of the *MR*-feature table are available as follows.

- *NewRecord(new_f, F)*: this creates and returns a new record of an *MR*-feature for a newly created form feature using the *MR*-feature list, $F = \{F_i\}_{i=0}^{n}$, and the database of form features. The new record is initially located to the last place (i.e., the $(n+1)$-th place).
- *DeleteRecord(F, j)*: this deletes the j-th record from the *MR*-feature list.
- *MoveRecord(F, j, m)*: this moves the j-th record to the m-th place. But, this operation does not change the definition of the effective zone, which is redefined by the upper-level function calling this function.

The algorithm for *MR*-feature modification consists of two steps, as shown in Algorithm 1. The system first searches for the destination place of the modified feature, and then moves the *MR*-feature to the destination place, as described in Algorithm 2. In this algorithm, the system moves the record of the *MR*-feature for the feature, F_j, to the m-th place first, and then redefines the effective zones of the features whose locations are changed. The effective zone of a feature defined by Eq. (7) is implemented in Algorithm 3. The definition of the effective zone, Z_j^i, of the feature, F_j^i, is represented as a string, and this string is recorded in the field of the record for the *MR*-feature. An example for feature modification is shown in Step 5 in Fig. 8.

Algorithm 1. ModifyMultiresolutionFeature (*F, j, m*)

```
Input: F: MR-feature list, F = {F_i}_{i=0}^n.
       j: current location of the MR-feature to be moved.
Output: F: rearranged MR-feature list.
        m: final location of the MR-feature F_j.
// Step 1. Find the destination location of the MR-feature F_j.
m ← FindLocation (F, j).
// Step 2. Move the j-th record to the m-th place in F and
   redefine the effective zones of the relocated features
   between F_j and F_m.
MoveMultiresolutionFeature(F, j, m).
```

Algorithm 2. MoveMultiresolutionFeature (*F, j, m*)

```
Input: F: MR-feature list, F = {F_i}_{i=0}^n.
   j: current location of the MR-feature to be moved.
   m: destination location of the MR-feature F_j.
Output: F: rearranged MR-feature list.
// Step 1. Move the j-th record to the m-th place in F
MoveRecord (F, j, m).
// Step 2. Redefine the effective zones of the relocated
   MR-features between F_j and F_m.
for k ← j to m do {
    RedefineEffectiveZone (F, k).
}
```

Algorithm 3. RedefineEffectiveZone (F, j)

1. **Input** F: MR-feature list, $F = \{F_i\}_{i=0}^{n}$.
2. j: location of the feature to be redefined in the rearranged order.
3. **Output**: F: rearranged feature list.
4. // $k(j)$: get the creation order i of F_j
5. $i \leftarrow$ CreationOrder of F_j.
6. // Initialize a character string S with the primitive $P_j (= P^i)$ of F_j.
7. $S \leftarrow P_j$.
8. **for** $\ell \leftarrow 0$ to $j-1$ **do** {
9. // $\alpha(j,\ell)$: check the type $\otimes_j$ of F_j and $\otimes_\ell$ of F_ℓ
10. **if** $\otimes_j \neq \otimes_\ell$ and ($\otimes_j = \cup$ or $\otimes_\ell = \cup$) **then** {
11. // $k(\ell)$: get the creation order k of F_ℓ
12. $k \leftarrow$ CreationOrder of F_ℓ.
13. // $\beta(i,k)$: check the precedence of the features
14. **if** $k < i$ **then** {
15. // $\underline{\otimes}(j,\ell)$: revise the effective zone of the feature F_j.
16. **if** $\otimes_j = \cup$ and $\otimes_\ell = \cap$ **then**
17. $S \leftarrow S + "\cap" + P_\ell$,
18. **else if** $\otimes_j = \cap$ and $\otimes_\ell = \cup$ **then**
19. $S \leftarrow S + "\cup" + P_\ell$,
20. **else**
21. $S \leftarrow S + "-" + P_\ell$.
22. }
23. }
24. }
25. // Set the effective zone Z_j of F_j with S.
26. $Z_j \leftarrow S$.

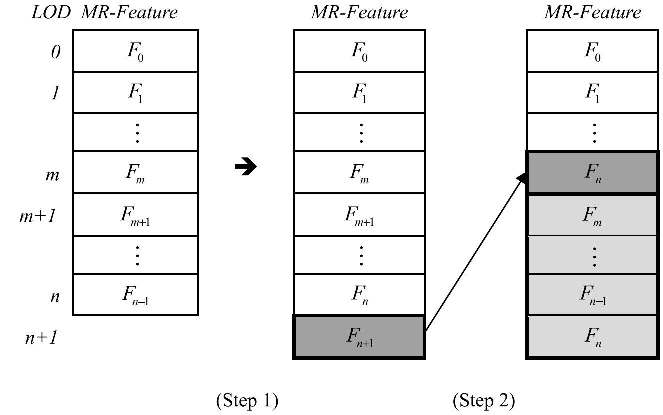

Fig. 14. Creation of a new *MR*-feature, followed by the rearrangement of the affected features: (Step 1) Create an *MR*-feature at the last place, (Step 2) Move the *MR*-feature according to the LOD criterion.

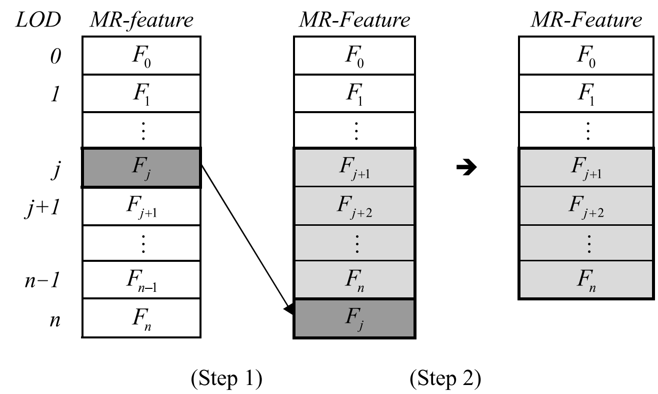

Fig. 15. Deletion of an *MR*-feature, followed by rearrangement of affected features: (*Step* 1) Move the *MR*-feature to be deleted to the last place, (*Step* 2) Delete the *MR*-feature.

Algorithm 4. CreateMultiresolutionFeature (*new_f, F, m*)

1. **Input**: *new_f*: new form feature.
2. F: MR-feature list, $F = \{F_i\}_{i=0}^{n}$.
3. **Output**: F: rearranged MR-feature list.
4. m: location of a new MR-feature.
5. // Step 1. Create a new record from the form feature *new_f* and get its location j ($= n+1$)
6. $j \leftarrow$ **NewRecord**(*new_f, F*):
7. // Step 2. Find the destination location of the feature F_j.
8. $m \leftarrow$ **FindLocation** (F, j).
9. // Step 3. Move the j-th record to the m-th place in F and redefine the effective zones of the relocated features between F_j and F_m.
10. **MoveMultiresolutionFeature**(F, j, m).

6.2.2. Feature creation

If a new form feature is created, the corresponding *MR*-feature is first created at the last place of the *MR*-feature list. Next, according to the LOD criterion, the adequate location of the *MR*-feature is investigated. Finally, the *MR*-feature is moved to that location. As illustrated in Fig. 14, the m-th to n-th *MR*-features are redefined due to the relocation of the new $n+1$-th *MR*-feature in Step 2. This feature creation algorithm is described in Algorithm 4. Compared to the feature modification algorithm, the feature creation algorithm has an additional preceding step for creating a new feature, followed by feature relocation. An example for this case is shown in Step 3 in Fig. 6.

6.2.3. Feature deletion

If a form feature is deleted, the system first relocates the corresponding *MR*-feature to the last place, and then deletes it, as illustrated in Fig. 15. The *MR*-features influenced by this deletion process are the j-th to n-th ones. The algorithm for feature deletion is described in Algorithm 5. Compared with the feature

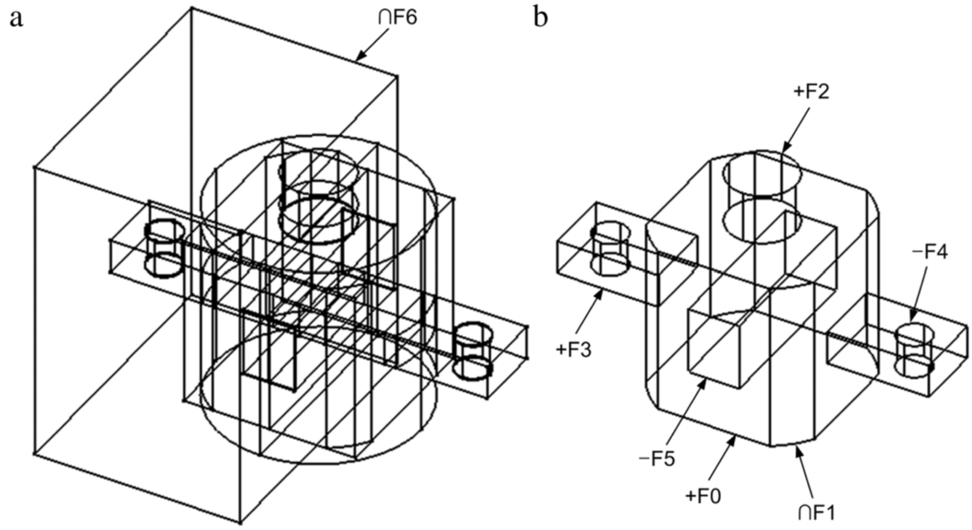

Fig. 16. An example part model: (a) a merged-set model for the part, (b) the final shape in feature-based design.

modification algorithm, this feature deletion algorithm has an additional step for deleting the *MR*-feature at the last place. An example of feature deletion is shown in Step 6 in Fig. 8.

Algorithm 5. DeleteMultiresolutionFeature (F, m)

1. **Input**: F: MR-feature list, $F = \{F_i\}_{i=0}^{n}$.
2. m: location of the MR-feature to be deleted.
3. **Output**: F: resultant MR-feature list.
4. // Step 1. Move the m-th record to the last n-th place in F and redefine the effective zones of the relocated features between F_m and F_n.
5. $n \leftarrow$ **GetLastLocation**(F).
6. **MoveMultiresolutionFeature**(F, m, n).
7. // Step 2. Delete the n-th record of F.
8. **DeleteRecord(**F, n):

7. Case study

7.1. Multiresolution modeling with intersective features

As mentioned in the introduction of this paper, in most CAD systems, Boolean operations including intersection are provided for the user together with feature modeling operations. For example, a sample part shown in Fig. 16 is created by applying five features, as shown in Fig. 17(a). Here, the third feature, F_3, is implemented using an intersection operation. If the volume of the feature is adopted as the LOD criterion, the LOD models in each step are displayed as shown in Fig. 17(b). Table 4 shows the *MR*-feature table used to define the LOD models in Step 6. This table contains a list of *MR*-features whose attributes include all the necessary information to extract the LOD models.

7.2. Effects of LOD criteria in multiresolution modeling

A simplified connecting rod was chosen as an example to study the effects of the LOD criteria in a simultaneous feature-based multiresolution modeling process. This part is modeled by applying 7 features, and the features are merged into a merged-set model, as shown in Fig. 18. Fig. 19 shows not only the feature-based modeling process but also a set of LOD models that are expandable for each modeling step. Here, the LOD criterion is the volume of the feature, regardless of whether the feature type is subtractive or additive. In Fig. 20, the LOD criterion is the volume of the subtractive feature. Here, the additive features are always precedent to the subtractive features. The lowest resolution model is obtained by uniting all the additive features, and the higher resolution models are obtained by subtracting the subtractive features in decreasing order of volume.

7.3. Multiresolution modeling for non-manifold topological models

The simultaneous feature-based multiresolution modeling technique proposed in this paper can be applied to a NMT model as well as a solid model. A simplified NMT plug part was chosen as an example for a case study. This part is modeled by applying six features, and the features are merged into a merged-set model, as shown in Fig. 21. Fig. 22(a) shows the feature-based modeling process, and Fig. 22(b) shows a set of LOD models that are expandable in each modeling step. Here, the area of the feature is adopted as the LOD criterion.

8. Conclusions

This paper introduced a new approach of incremental feature-based multiresolution modeling simultaneously executable with feature-based part modeling in product design. In this approach, the feature-based multiresolution modeling task is performed simultaneously with feature-based part modeling, so that the LOD models for the current stage of the CAD model can be provided immediately upon the designer's request. With this feature, the designer can carry out the product design task more efficiently by utilizing the multiresolution models for various applications including fast rendering, network-based collaborative design, and engineering analyses.

To implement this approach, we developed the history-based selective Boolean (*HS*-Boolean) operations, in which the order of the *HS*-Boolean operations of union, difference, and intersection may alter, unlike the conventional Boolean operations, by redefining the region affected by the operation. Because of this property, even if the features are rearranged arbitrarily regardless of the feature type, feature-based multiresolution modeling based

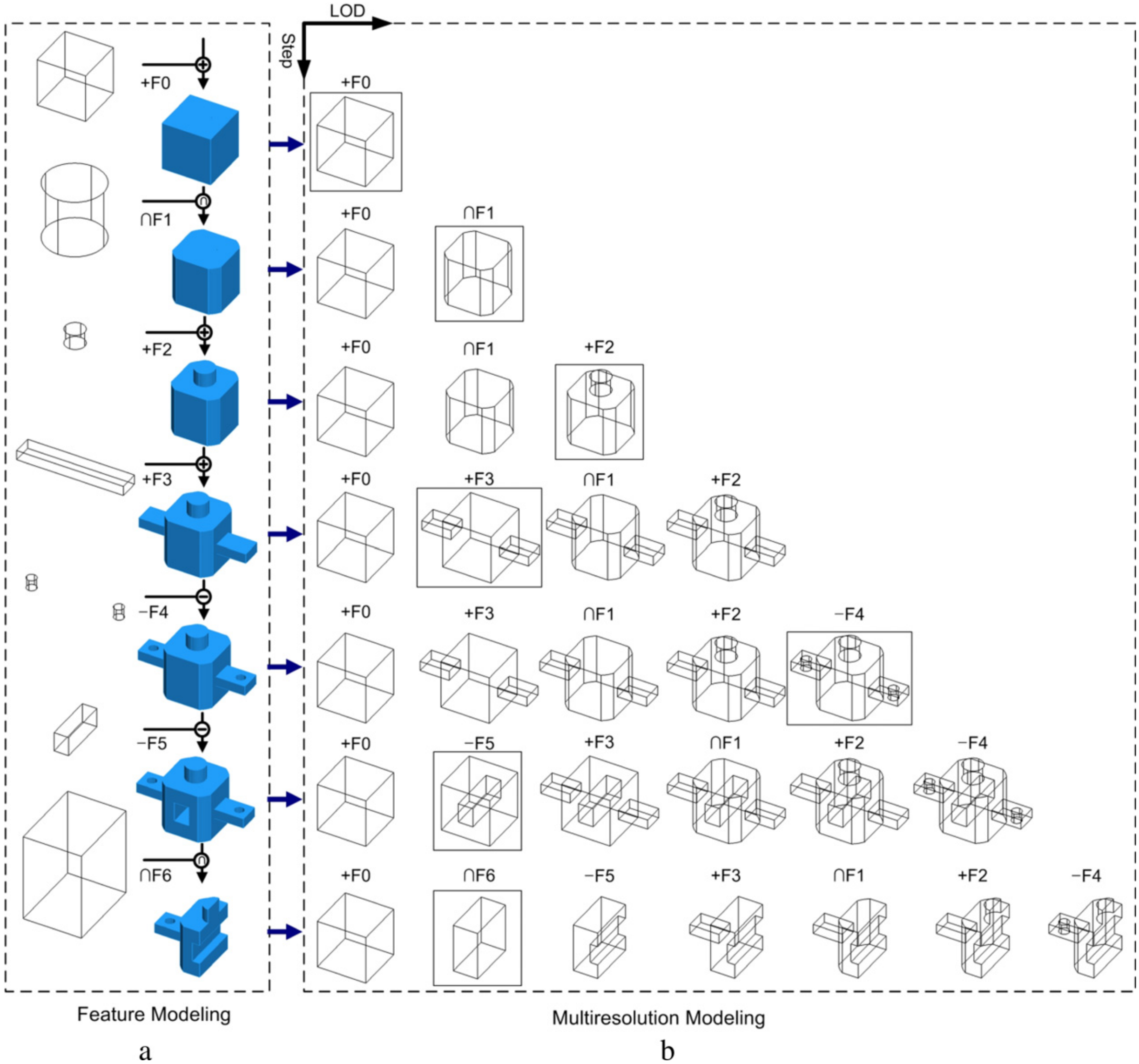

Fig. 17. Simultaneous feature-based multiresolution modeling process for the sample model shown in Fig. 16: (a) feature-based solid modeling process, (b) LOD model at each step of the feature-based modeling process, where the LOD criterion is the volume of the feature.

Table 4
MR-feature table in Step 4.

No. (LOD)	Feature name	Creation order	Bool	Primitive	Effective zone	LOD model
0	Base	0	+	$P0$	$P0$	$P0$
1	Intersect_Block	6	$*$	$P6$	$P6$	$P0 * P6$
2	Rect_Hole	5	−	$P5$	$P5$	$P0 * P6 - P5$
3	Rod	3	+	$P3$	$P3 * P6 - P5$	$P0 * P6 - P5 + (P3 * P6 - P5)$
4	Intersect_Cyl	1	$*$	$P1$	$P1 + P3$	$P0 * P6 - P5 + (P3 * P6 - P5) * (P1 + P3)$
5	Boss	2	+	$P2$	$P2 * P6 - P5$	$P0 * P6 - P5 + (P3 * P6 - P5) * (P1 + P3) + (P2 * P6 - P5)$
6	HolePattern	4	−	$P4$	$P4$	$P0 * P6 - P5 + (P3 * P6 - P5) * (P1 + P3) + (P2 * P6 - P5) - P4$

on these operations can guarantee the same part shape as the original, and provide a valid and unique shape at each intermediate LOD. In addition, the *HS*-Boolean operations can be implemented using conventional solid data structures as well as NMT data structures because these operations are independent of the topological framework.

This paper also presented an interesting finding that the result of a set of the *HS*-Boolean operations in an arbitrary order is always the same as the result of the same set of Boolean operations in the original order of creation. Using this property, it is possible to obtain an LOD model by suppressing the irrelevant features from the feature tree without changing the order of the features. This approach allows easy implementation of feature-based multiresolution modeling in current commercial CAD systems because transition features do not have to be converted to volume features.

To validate this approach, we implemented a prototype system based on an NMT modeling kernel. The *HS*-Boolean operations were implemented based on the merge-and-select algorithm and applied to the development of the feature-based multiresolution modeling module. Although this approach allows for the fast extraction of multiresolution models for given LODs, it still

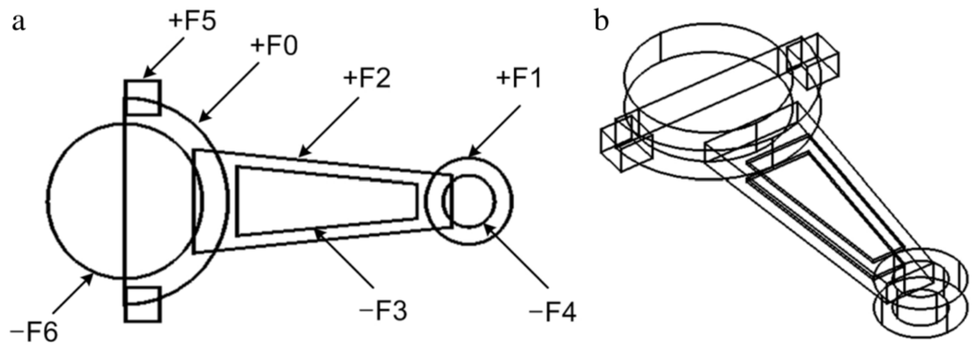

Fig. 18. A simplified connecting rod for case study: (a) seven features constituting the part. (b) A merged-set model for the part.

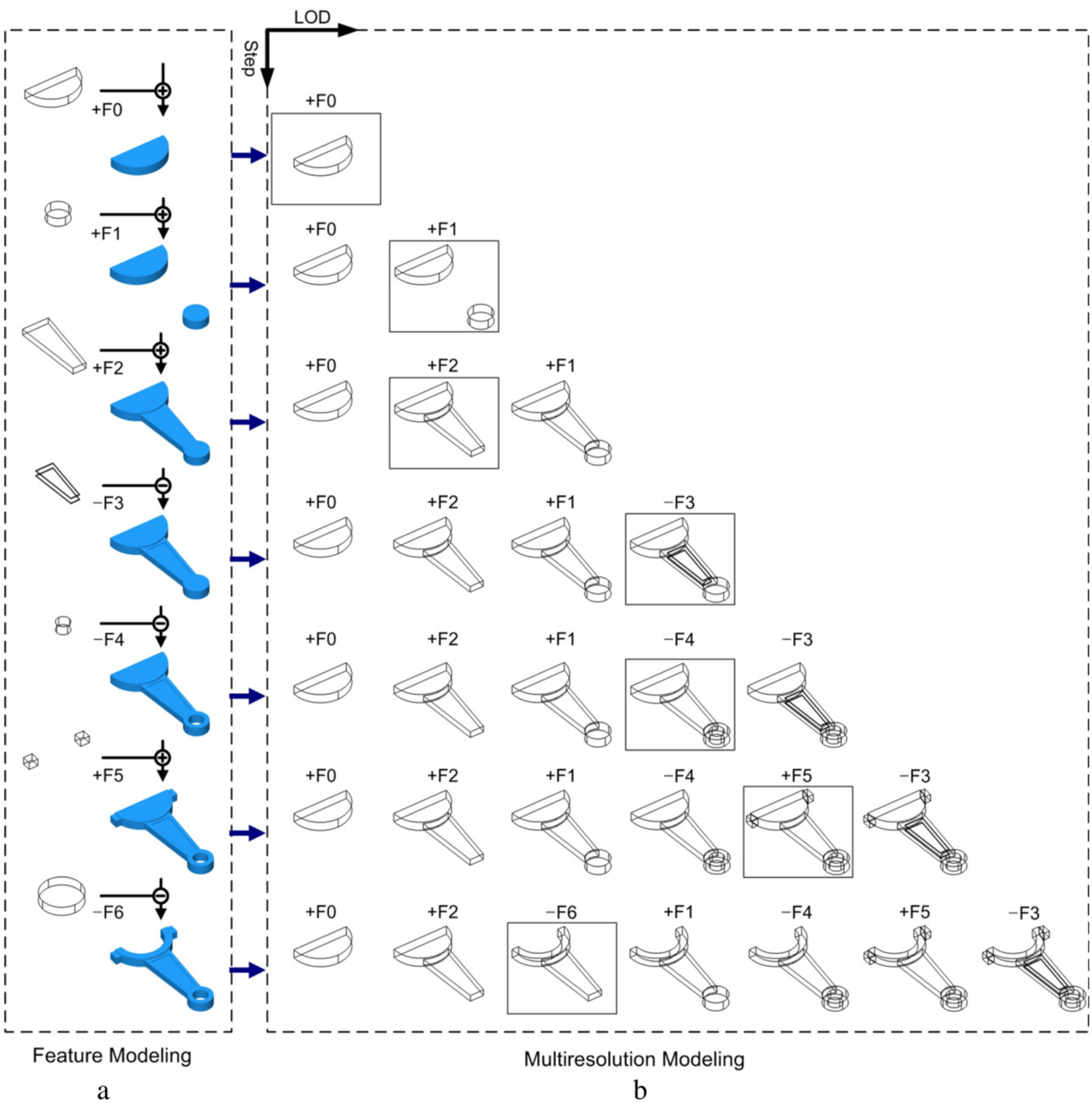

Fig. 19. Simultaneous feature-based multiresolution modeling process for a connecting rod: (a) feature-based solid modeling process, (b) LOD models at each step of the feature-based design process, where the LOD criterion is the volume of the feature, regardless of the feature type.

requires a lot of computation time for a large CAD model when building a NMT merge set model.

Our approach is able to support not only volume features but also transition features, such as fillet or draft. However, since our

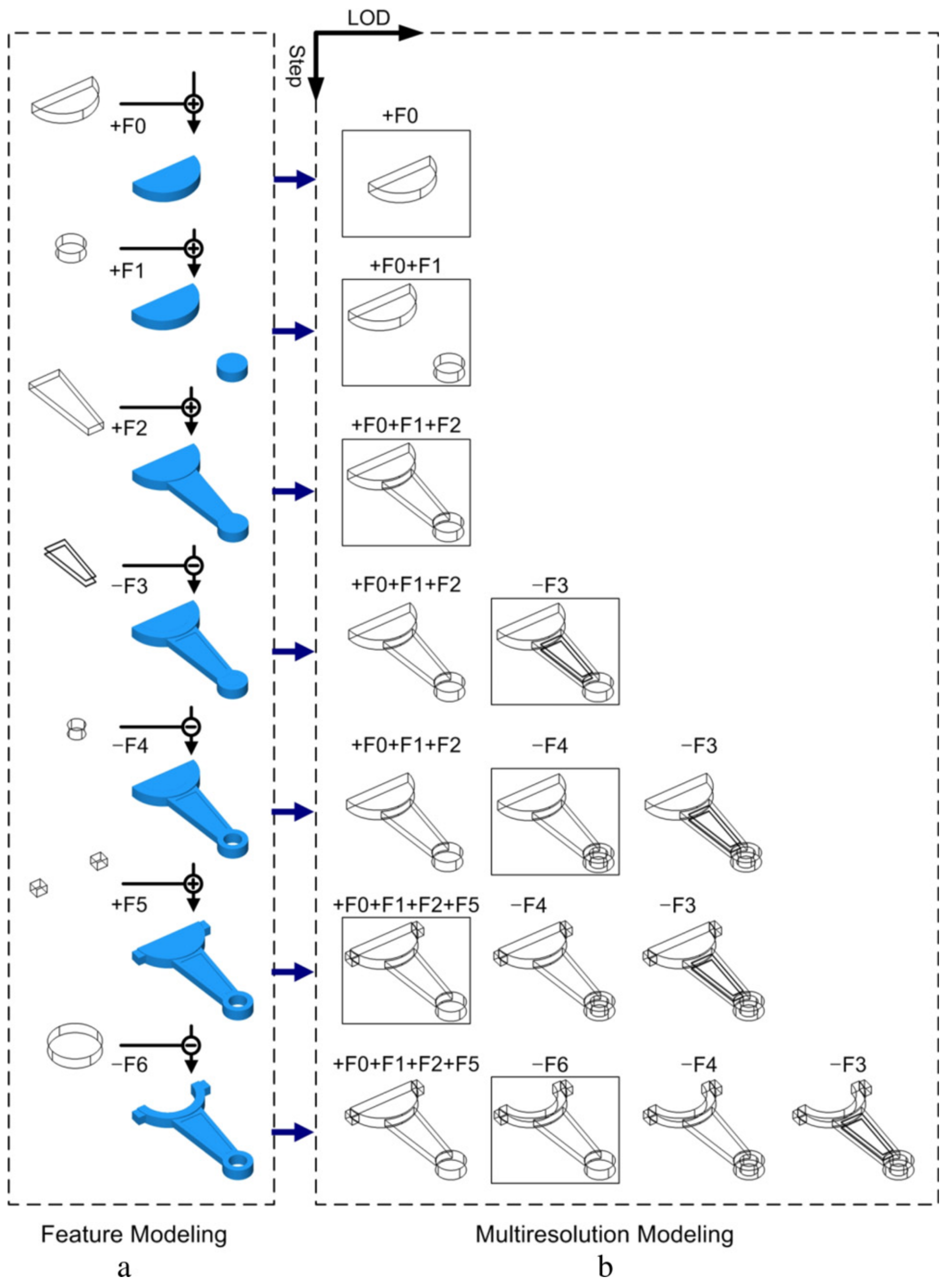

Fig. 20. Simultaneous feature-based multiresolution modeling process for a connecting rod: (a) feature-based solid modeling process, (b) LOD models at each step of the feature-based modeling process, where the LOD criterion is the volume of the subtractive feature.

approach basically relies on the representation of volume features, transition features are converted to volume features, i.e., delta volumes, which are obtained by performing Boolean operations between the current and previous steps of the part model. This work is computationally expensive and error-prone. This problem can be overcome if the cellular representation is adopted like our prototype system, and those Boolean operations are replaced with topological inquiry functions that search for delta volumes by traveling the topological entities of the cellular model from the newly generated faces of the transition feature.

To complete the feature-based multiresolution modeling technology, future work still remains on the following topics: fast multiresolution modeling techniques for practical use, in particular, for large CAD models; feature-based multiresolution modeling of assembly models, which is useful for virtual prototyping and manufacturing as well as engineering design; more robust and seamless integration of CAD and CAE systems based on this technique; exploring diverse application areas including engineering analysis, visualization, network-based collaborative design, and so on.

Acknowledgments

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-313-D00066). It was also supported by the Korea Science and Engineering Foundation (KOSEF) Grant funded by the Korea government (MOST) (No. R11-2007-028-02001-0). The first author's work was supported by Research Program 2011 of Kookmin University in Korea as well.

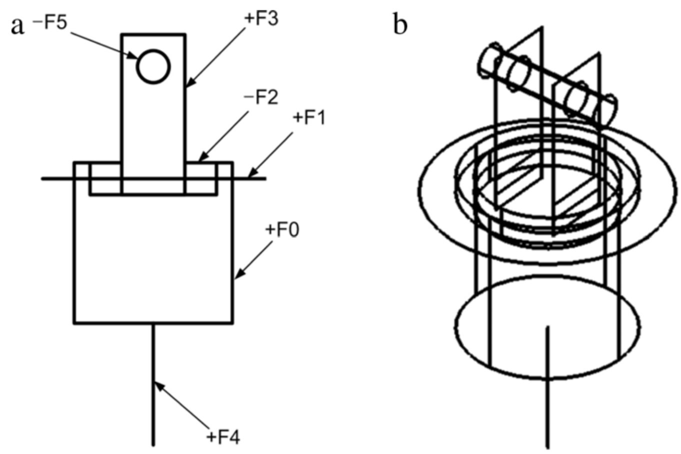

Fig. 21. A simplified NMT model for a plug: (a) the features constituting the part. (b) A merged-set model for the part.

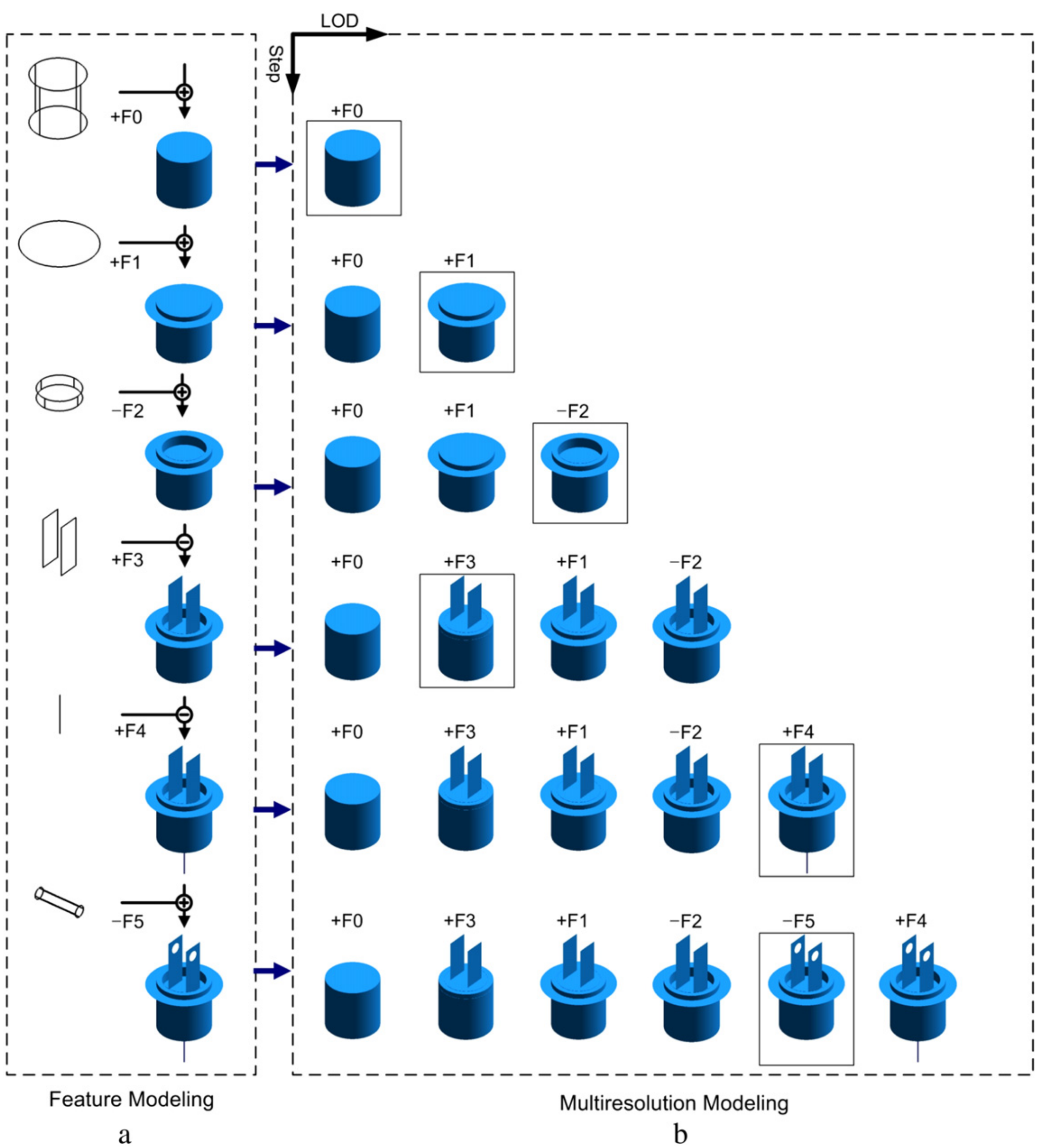

Fig. 22. Simultaneous feature-based multiresolution modeling process for a plug: (a) feature-based solid modeling process, (b) LOD model at each step of the feature-based modeling process, where the LOD criterion is the area of the feature.

Appendix A. Exchange of two Boolean operations (proof of Eq. (9))

The detailed proofs of Cases 5–9 in Table 1 are described in this section. If $\overline{P}^1$ denotes the complement of P^1, the difference operation can be represented by

$$P^0 - P^1 = P^0 \cap \overline{P}^1 \tag{A.1}$$

as introduced in Ref. [38].

(*Case* 5) $P^0 \cap P^1 \cap P^2 = P^0 \cap P^2 \cap P^1$.

$$\begin{aligned} P^0 \cap P^1 \cap P^2 &= (P^0 \cap P^1) \cap P^2 \\ &= P^0 \cap (P^1 \cap P^2) \quad \text{(associative law)} \\ &= P^0 \cap (P^2 \cap P^1) \quad \text{(commutative law)} \\ &= (P^0 \cap P^2) \cap P^1 \quad \text{(associative law)} \\ &= P^0 \cap P^2 \cap P^1. \end{aligned} \tag{A.2}$$

(*Case* 6) $P^0 \cup P^1 \cap P^2 = P^0 \cap P^2 \cup (P^1 \cap P^2)$.

$$\begin{aligned} P^0 \cup P^1 \cap P^2 &= (P^0 \cup P^1) \cap P^2 \\ &= (P^0 \cup P^1) - \overline{P}^2 \quad \text{(Eq. (A.1))} \\ &= (P^0 - \overline{P}^2) \cup (P^1 - \overline{P}^2) \quad \text{(Case 3)} \\ &= (P^0 \cap P^2) \cup (P^1 \cap P^2) \quad \text{(Eq. (A.1))} \\ &= P^0 \cap P^2 \cup (P^1 \cap P^2). \end{aligned} \tag{A.3}$$

(*Case* 7) $P^0 \cap P^1 \cup P^2 = P^0 \cup P^2 \cap (P^1 \cup P^2)$.

$$\begin{aligned} P^0 \cap P^1 \cup P^2 &= P^0 \cup P^2 \cap (P^1 \cup P^2) \\ &= (P^0 - \overline{P}^1) \cup P^2 \quad \text{(Eq. (A.1))} \\ &= (P^0 \cup P^2) - (\overline{P^1} - P^2) \quad \text{(Case 4)} \\ &= (P^0 \cup P^2) - (\overline{P^1} \cap \overline{P^2}) \quad \text{(Eq. (A.1))} \\ &= (P^0 \cup P^2) - \overline{(P^1 \cup P^2)} \quad \text{(De Morgan's law)} \\ &= (P^0 \cup P^2) \cap (P^1 \cup P^2) \quad \text{(Eq. (A.1))} \\ &= P^0 \cup P^2 \cap (P^1 \cup P^2). \end{aligned} \tag{A.4}$$

(*Case* 8) $P^0 - P^1 \cap P^2 = P^0 \cap P^1 - P^2$.

$$\begin{aligned} P^0 - P^1 \cap P^2 &= (P^0 - P^1) \cap P^2 \\ &= (P^0 \cap \overline{P^1}) \cap P^2 \quad \text{(Eq. (A.1))} \\ &= P^0 \cap (\overline{P^1} \cap P^2) \quad \text{(associative law)} \\ &= P^0 \cap (P^2 \cap \overline{P^1}) \quad \text{(commutative law)} \\ &= (P^0 \cap P^2) \cap \overline{P^1} \quad \text{(associative law)} \\ &= (P^0 \cap P^2) - P^1 \quad \text{(Eq. (A.1))} \\ &= P^0 \cap P^2 - P^1. \end{aligned} \tag{A.5}$$

(*Case* 9) $P^0 \cap P^1 - P^2 = P^0 - P^2 \cap P^1$.

$$\begin{aligned} P^0 \cap P^1 - P^2 &= (P^0 \cap P^1) - P^2 \\ &= (P^0 \cap P^1) \cap \overline{P^2} \quad \text{(Eq. (A.1))} \\ &= P^0 \cap (P^1 \cap \overline{P^2}) \quad \text{(associative law)} \\ &= P^0 \cap (\overline{P^2} \cap P^1) \quad \text{(commutative law)} \\ &= (P^0 \cap \overline{P^2}) \cap P^1 \quad \text{(associative law)} \\ &= (P^0 - P^2) \cap P^1 \quad \text{(Eq. (A.1))} \\ &= P^0 - P^2 \cap P^1. \end{aligned} \tag{A.6}$$

Appendix B. Relocation of a single *HS*-Boolean operation (proof of Eq. (10))

Let M_n denote the resulting model obtained by applying n Boolean operations between $n+1$ models:

$$M_n = \prod_{k=0}^{n} \otimes^k P^k, \quad \text{where } \otimes^0 P^0 = \varnothing \otimes^0 P^0. \tag{B.1}$$

Let us investigate the case in which the i-th Boolean operation $\otimes^i P^i$ is moved to the j-th position and $i < j$. This movement can be decomposed into a series of exchanges between two adjacent Boolean operations. Each exchange is examined as follows until the i-th operation arrives at the j-th place.

(*Step* 1) Investigating the first step in which $\otimes^i P^i$ is replaced with $\otimes^{i+1} P^{i+1}$, the initial model before the replacement, M_n, can be represented as

$$M_n = M_{i+1} \prod_{k=i+2}^{n} \otimes^k P^k \tag{B.2}$$

where

$$M_{i+1} = M_{i-1} \otimes^i P^i \otimes^{i+1} P^{i+1}. \tag{B.3}$$

If the j-th and $j+1$-th elements in Eq. (B.3) are exchanged with each other, then

$$\begin{aligned} \overset{(1)}{M}_{i+1} &= M_{i+1} \\ &= M_{i-1} \otimes^{i+1} P^{i+1} \otimes^i (P^i \underline{\otimes} (i, i+1) \alpha(i, i+1) P^{i+1}). \end{aligned} \tag{B.4}$$

Therefore, the model after the exchange, M_n', can be rewritten as

$$\begin{aligned} \overset{(1)}{M}_n &= M_n = \prod_{k=0}^{n} \overset{(1)}{\otimes}{}^k \overset{(1)}{P}{}^k \\ &= \left(\prod_{k=0}^{i-1} \otimes^k P^k \right) \otimes^{i+1} P^{i+1} \\ &\quad \times \otimes^i \left(P^i \underline{\otimes} (i, i+1) \alpha(i, i+1) P^{i+1}\right) \left(\prod_{k=i+2}^{n} \otimes^k P^k \right). \end{aligned} \tag{B.5}$$

(*Step* 2) Investigating the next step, in which $\overset{(1)}{\otimes}{}^{i+1} \overset{(1)}{P}{}^{i+1}$ and $\overset{(1)}{\otimes}{}^{i+2} \overset{(1)}{P}{}^{i+2}$ are exchanged with each other, the model before the exchange, $\overset{(1)}{M}_n$, can be written as

$$\overset{(1)}{M}_n = \overset{(1)}{M}_{i+2} \prod_{k=i+3}^{n} \overset{(1)}{\otimes}{}^k \overset{(1)}{P}{}^k = \overset{(1)}{M}_{i+2} \prod_{k=i+3}^{n} \otimes^k P^k \tag{B.6}$$

where

$$\overset{(1)}{M}_{i+2} = \overset{(1)}{M}_i \overset{(1)}{\otimes}{}^{i+1} \overset{(1)}{P}{}^{i+1} \overset{(1)}{\otimes}{}^{i+2} \overset{(1)}{P}{}^{i+2}. \tag{B.7}$$

If $i+1$-th and $i+2$-th elements of Eq. (B.7) are exchanged with each other, then

$$\begin{aligned} \overset{(2)}{M}_{i+2} &= \overset{(1)}{M}_{i+2} = \overset{(1)}{M}_i \overset{(1)}{\otimes}{}^{i+2} \overset{(1)}{P}{}^{i+2} \\ &\quad \times \overset{(1)}{\otimes}{}^{i+1} (\overset{(1)}{P}{}^{i+1} \underline{\overset{(1)}{\otimes}}(i+1, i+2) \overset{(1)}{\alpha}(i+1, i+2) \overset{(1)}{P}{}^{i+2}). \end{aligned} \tag{B.8}$$

Using Eq. (B.4), this equation can be rewritten as

$$\begin{aligned} \overset{(2)}{M}_{i+1} &= M_{i-1} \otimes^{i+1} P^{i+1} \otimes^{i+2} P^{i+2} \\ &\quad \times \otimes^i (P^i \underline{\otimes} (i, i+1) \alpha(i, i+1) P^{i+1} \\ &\quad \times \underline{\otimes} (i, i+2) \alpha(i, i+2) P^{i+2}). \end{aligned} \tag{B.9}$$

Therefore, the model after the exchange, $\overset{(2)}{M}_n$, is represented as

$$\begin{aligned}\overset{(2)}{M}_n &= \overset{(1)}{M}_n = M_n = \prod_{k=0}^{n} \overset{(2)}{\otimes}{}^k \overset{(2)}{P}{}^k \\ &= \left(\prod_{k=0}^{i-1} \otimes^k P^k\right) \otimes^{i+1} P^{i+1} \otimes^{i+2} P^{i+2} \\ &\quad \times \otimes_i \left(P^i \underline{\otimes}\,(i, i+1)\alpha(i, i+1)P^{i+1}\right. \\ &\quad \times \left.\underline{\otimes}\,(i, i+2)\alpha(i, i+2)P^{i+2}\right) \left(\prod_{k=i+3}^{n} \otimes^k P^k\right) \\ &= \left(\prod_{k=0}^{i-1} \otimes^k P^k\right) \left(\prod_{k=i+1}^{i+2} \otimes^k P^k\right) \\ &\quad \times \otimes_i \left(P^i \underline{\otimes}\,(i, i+1)\alpha(i, i+1)P^{i+1}\right. \\ &\quad \times \left.\underline{\otimes}\,(i, i+2)\alpha(i, i+2)P^{i+2}\right) \left(\prod_{k=i+3}^{n} \otimes^k P^k\right). \end{aligned} \tag{B.10}$$

(*Step m*) Investigating the m-th step, in which the $i+m-1$-th and $i+m$-th elements are exchanged with each other, the resultant model from the $m-1$ exchanges, $\overset{(m-1)}{M_n}$, can be represented as

$$\overset{(m-1)}{M_n} = \overset{(m-1)}{M}_{i+m} \prod_{k=i+m+1}^{n} \overset{(m-1)}{\otimes^k} \overset{(m-1)}{P^k} \tag{B.11}$$

where

$$\overset{(m-1)}{M}_{i+m} = \overset{(m-1)}{M}_{i+m-2} \overset{(m-1)}{\otimes}{}^{i+m-1} \overset{(m-1)}{P}{}^{i+m-1} \overset{(m-1)}{\otimes}{}^{i+m} \overset{(m-1)}{P}{}^{i+m}. \tag{B.12}$$

If the $i+m-1$-th and $i+m$-th elements of Eq. (B.12) are exchanged with each other, the model $\overset{(m)}{M}_{i+m}$ can be described as follows.

$$\begin{aligned}\overset{(m)}{M}_{i+m} &= \overset{(m-1)}{M}_{i+m} = \overset{(m-1)}{M}_{i+m-2} \overset{(m-1)}{\otimes}{}^{i+m} \overset{(m-1)}{P}{}^{i+m} \\ &\quad \times \overset{(m-1)}{\otimes}{}^{i+m-1}(\overset{(m-1)}{P}{}^{i+m-1} \overset{(m-1)}{\underline{\otimes}}(i+m-1, i+m) \\ &\quad \times \overset{(m-1)}{\alpha}(i+m-1, i+m) \overset{(m-1)}{P}{}^{i+m}). \end{aligned} \tag{B.13}$$

Using the initial elements, this formula can be rewritten as

$$\begin{aligned}\overset{(m)}{M}_{i+m} &= \left(\prod_{k=0}^{i-1} \otimes^k P^k\right) \left(\prod_{k=i+1}^{i+m} \otimes^k P^k\right) \\ &\quad \times \otimes^i \left(P^i \prod_{\ell=1}^{m} \underline{\otimes}\,(i, i+\ell)\alpha(i, i+\ell)P^{i+\ell}\right). \end{aligned} \tag{B.14}$$

Therefore, the model after the m exchanges, $\overset{(m)}{M}_n$, can be written as

$$\begin{aligned}\overset{(m)}{M}_n &= M_n = \prod_{k=0}^{n} \overset{(m)}{\otimes}{}^k \overset{(m)}{P}{}^k = \left(\prod_{k=0}^{i-1} \otimes^k P^k\right) \left(\prod_{k=i+1}^{i+m} \otimes^k P^k\right) \\ &\quad \times \otimes^i \left(P^i \prod_{\ell=1}^{m} \underline{\otimes}(i, i+\ell)\alpha(i, i+\ell)P^{i+\ell}\right) \\ &\quad \times \left(\prod_{k=i+m+1}^{n} \otimes^k P^k\right). \end{aligned} \tag{B.15}$$

When the i-th element is moved to the j-th location, $m = j - i$. If $m = j - i$ is applied to Eq. (B.15), the following formula is obtained, and this is equal to Eq. (12).

$$\begin{aligned}M_n &= \left(\prod_{k=0, k\neq i}^{j} \otimes^k P^k\right) \otimes^i \left(P^i \prod_{\ell=1}^{j-i} \underline{\otimes}(i, i+\ell)\alpha(i, i+\ell)P^{i+\ell}\right) \\ &\quad \times \left(\prod_{k=j+1}^{n} \otimes^k P^k\right). \end{aligned} \tag{B.16}$$

Appendix C. Rearrangement of all the *HS*-Boolean operations (proof of Eq. (11))

Let us follow the feature rearrangement algorithm proposed by Lee [5], in which the features are relocated one by one, starting with the least significant feature, according to the feature significance as follows. The features are initially arranged in the order of creation. First, the least significant feature is selected, and moved to the n-th place. Next, the second least significant feature is selected, and moved to the $n-1$-th place. This movement process is repeated until the most significant feature is located at the 0-th place. Whenever each feature is moved to its new place, the new effective zone is redefined using Eq. (10). Let us follow this algorithm to rearrange the *HS*-Boolean operations. If $\overset{(k)}{P_j}$ denotes the primitive of an *HS*-Boolean operation, which is located at the j-th position in the k-th moving step, the effective zone in each step can be derived as follows. Table C.1 summarizes this rearrangement process.

(*Step* 1) Let us assume that the first selected operation is initially located at the a-th place, $(0 \le a \le n)$, and its primitive is P^a. If the operation is moved to n-th place, then its effective zone, Z_n^a, can be obtained from the following formula defined by the terms for the initial operation order, for the step-1 order, and for the final step-n order, respectively. For the step-1 and step-n order, $\underline{\otimes}(n, \ell)$ is determined according to $\otimes_n$: if $\otimes_n = \cup$, $\underline{\otimes}(n, \ell) = \cap$ or $-$; if $\otimes_n = \cap$, $\underline{\otimes}(n, \ell) = \cup$; if $\otimes_n = -$, $\underline{\otimes}(n, \ell) = -$. Since $\cap$ and $-$ are exchangeable, Eq. (C.1) can be derived.

$$\begin{aligned}Z_n^a &= \overset{(0)}{P}{}_a^a \prod_{\ell=a+1}^{n} \overset{(0)}{\underline{\otimes}}(a, \ell)\, \overset{(0)}{\alpha}(a, \ell)\, \overset{(0)}{P}_\ell \\ &= \overset{(1)}{P}{}_n^a \prod_{\ell=a}^{n-1} \overset{(1)}{\underline{\otimes}}(n, \ell)\, \overset{(1)}{\alpha}(n, \ell)\, \overset{(1)}{P}_\ell \\ &= P_n^a \prod_{\ell=0}^{n-1} \underline{\otimes}(n, \ell)\alpha(n, \ell)\, \beta(a, k(\ell))P_\ell^{k(\ell)}. \end{aligned} \tag{C.1}$$

(*Step* 2) Let us assume that the second selected operation is located at an arbitrary x-th place, $(0 \le x \le n-1)$, for the step-1 order, and its primitive is P^b. If the operation is moved to the $n-1$-th place, then its effective zone becomes

$$\begin{aligned}Z_{n-1}^b &= \overset{(1)}{P}{}_x^b \prod_{\ell=x+1}^{n-1} \overset{(1)}{\underline{\otimes}}(x, \ell)\, \overset{(1)}{\alpha}(x, \ell)\, \overset{(1)}{P}_\ell \\ &= \overset{(2)}{P}{}_{n-1}^b \prod_{\ell=x}^{n-2} \overset{(2)}{\underline{\otimes}}(x, \ell)\, \overset{(2)}{\alpha}(x, \ell)\, \overset{(2)}{P}_\ell \\ &= P_{n-1}^b \prod_{\ell=0}^{n-2} \underline{\otimes}(n-1, \ell)\alpha(n-1, \ell)\, \beta(b, k(\ell))P_\ell^{k(\ell)}. \end{aligned} \tag{C.2}$$

(*Step m*) Let us assume that the m-th selected operation is located at an arbitrary y-th place, $(0 \le y \le n-m+1)$, for the step-$(m-1)$ order, and its primitive is P^i. If the operation is moved to

Table C.1
Rearrangement of the *HS*-Boolean operations according to the feature rearrangement algorithm, in which features are relocated starting with the least significant feature according to feature significance.

Step	Relocation (original → current)	0	1	2	⋯	y	⋯	$n-m$	$n-m+1$	⋯	a	⋯	x	⋯	$n-2$	$n-1$	n
0		$\overset{(0)}{P}_0$	$\overset{(0)}{P}_1$	$\overset{(0)}{P}_2$	⋯	$\overset{(0)}{P}_y$	⋯	$\overset{(0)}{P}_{n-m}$	$\overset{(0)}{P}_{n-m+1}$	⋯	$\overset{(0)}{P}_a$	⋯	$\overset{(0)}{P}_x$	⋯	$\overset{(0)}{P}_{n-2}$	$\overset{(0)}{P}_{n-1}$	$\overset{(0)}{P}_n$
		P^0	P^1	P^2	⋯	P^y	⋯	P^{n-m}	P^{n-m+1}	⋯	P^a	⋯	P^x	⋯	P^{n-2}	P^{n-1}	P^n
1	$a \to n$	$\overset{(0)}{P}_0$	$\overset{(0)}{P}_1$	$\overset{(0)}{P}_2$	⋯	$\overset{(0)}{P}_y$	⋯	$\overset{(0)}{P}_{n-m}$	$\overset{(0)}{P}_{n-m+1}$	⋯	$\overset{(0)}{P}_{a+1}$	⋯	$\overset{(0)}{P}_{x+1}$	⋯	$\overset{(0)}{P}_{n-1}$	$\overset{(0)}{P}_n$	Z_n^a
		$\overset{(1)}{P}_0$	$\overset{(1)}{P}_1$	$\overset{(1)}{P}_2$	⋯	$\overset{(1)}{P}_y$	⋯	$\overset{(1)}{P}_{n-m}$	$\overset{(1)}{P}_{n-m+1}$	⋯	$\overset{(1)}{P}_a$	⋯	$\overset{(1)}{P}_x (= P^b)$	⋯	$\overset{(1)}{P}_{n-2}$	$\overset{(1)}{P}_{n-1}$	Z_n^a
2	$b \to n-1$	$\overset{(1)}{P}_0$	$\overset{(1)}{P}_1$	$\overset{(1)}{P}_2$	⋯	$\overset{(1)}{P}_y$	⋯	$\overset{(1)}{P}_{n-m}$	$\overset{(1)}{P}_{n-m+1}$	⋯	$\overset{(1)}{P}_a$	⋯	$\overset{(1)}{P}_{x+1}$	⋯	$\overset{(1)}{P}_{n-1}$	Z_{n-1}^b	Z_n^a
		$\overset{(2)}{P}_0$	$\overset{(2)}{P}_1$	$\overset{(2)}{P}_2$	⋯	$\overset{(2)}{P}_y$	⋯	$\overset{(2)}{P}_{n-m}$	$\overset{(2)}{P}_{n-m+1}$	⋯	$\overset{(2)}{P}_a$	⋯	$\overset{(2)}{P}_x$	⋯	$\overset{(2)}{P}_{n-1}$	Z_{n-1}^b	Z_n^a
⋮	⋮	⋮	⋮	⋮		⋮		⋮	⋮		⋮		⋮		⋮	⋮	⋮
$m-1$		$\overset{(m-1)}{P}_0$	$\overset{(m-1)}{P}_1$	$\overset{(m-1)}{P}_2$	⋯	$\overset{(m-1)}{P}_y (= P^i)$	⋯	$\overset{(m-1)}{P}_{n-m}$	$\overset{(m-1)}{P}_{n-m+1}$	⋯	$Z_a^{k(a)}$	⋯	$Z_x^{k(x)}$	⋯	$Z_{n-2}^{k(n-2)}$	Z_{n-1}^b	Z_n^a
m	$i \to n-m+1$	$\overset{(m-1)}{P}_0$	$\overset{(m-1)}{P}_1$	$\overset{(m-1)}{P}_2$	⋯	$\overset{(m-1)}{P}_{y+1}$	⋯	$\overset{(m-1)}{P}_{n-m+1}$	Z_{n-m+1}^i	⋯	$Z_a^{k(a)}$	⋯	$Z_x^{k(x)}$	⋯	$Z_{n-2}^{k(n-2)}$	Z_{n-1}^b	Z_n^a
		$\overset{(m)}{P}_0$	$\overset{(m)}{P}_1$	$\overset{(m)}{P}_2$	⋯	$\overset{(m)}{P}_y$	⋯	$\overset{(m)}{P}_{n-m}$	Z_{n-m+1}^i	⋯	$Z_a^{k(a)}$	⋯	$Z_x^{k(x)}$	⋯	$Z_{n-2}^{k(n-2)}$	Z_{n-1}^b	Z_n^a
⋮	⋮	⋮	⋮	⋮		⋮		⋮	⋮		⋮		⋮		⋮	⋮	⋮
n	$s \to 1$	$\overset{(n-1)}{P}_1$	Z_1^s	$Z_2^{k(2)}$	⋯	$Z_y^{k(y)}$	⋯	$Z_{n-m}^{k(n-m)}$	Z_{n-m+1}^i	⋯	$Z_a^{k(a)}$	⋯	$Z_x^{k(x)}$	⋯	$Z_{n-2}^{k(n-2)}$	Z_{n-1}^b	Z_n^a
		$\overset{(n)}{P}_0$	Z_1^s	$Z_2^{k(2)}$	⋯	$Z_y^{k(y)}$	⋯	$Z_{n-m}^{k(n-m)}$	Z_{n-m+1}^i	⋯	$Z_a^{k(a)}$	⋯	$Z_x^{k(x)}$	⋯	$Z_{n-2}^{k(n-2)}$	Z_{n-1}^b	Z_n^a
Final effective zone		$Z_0^{k(0)}$	$Z_1^{k(1)}$	$Z_2^{k(2)}$	⋯	$Z_y^{k(y)}$	⋯	$Z_{n-m}^{k(n-m)}$	$Z_{n-m+1}^{k(n-m+1)}$	⋯	$Z_a^{k(a)}$	⋯	$Z_x^{k(x)}$	⋯	$Z_{n-2}^{k(n-2)}$	$Z_{n-2}^{k(n-1)}$	$Z_{n-2}^{k(n)}$

the $n-m+1$-th place, then its effective zone becomes

$$\begin{aligned} Z^i_{n-m+1} &= \overset{(m-1)}{P^i_y} \prod_{\ell=y+1}^{n-m+1} \overset{(m-1)}{\underline{\otimes}}(y,\ell) \overset{(m-1)}{\alpha}(y,\ell) \overset{(m-1)}{P_\ell} \\ &= \overset{(m)}{P^i_{n-m+1}} \prod_{\ell=y}^{n-m} \overset{(m)}{\underline{\otimes}}(y,\ell) \overset{(m)}{\alpha}(y,\ell) \overset{(m)}{P_\ell} \\ &= P^i_{n-m+1} \prod_{\ell=0}^{n-m} \underline{\otimes}(n-m+1,\ell)\alpha(n-m+1,\ell) \\ &\quad \times \beta(i,k(\ell))P^{k(\ell)}_\ell. \end{aligned} \tag{C.3}$$

When this process is repeated until Step n, all the effective zones of the rearranged operations are eventually determined. It should be noted that in Step m, P^i is moved to the j-th place for the step-$(m-1)$ order. Then, j becomes $j = n-m+1$. If $m = n-j+1$ and $P^i_j = P^i$ are applied to Eq. (C.3), then the effective zones of the rearranged operations become

$$Z^i_j = P^i_j \prod_{\ell=0}^{j-1} \underline{\otimes}(j,\ell)\alpha(j,\ell)\beta(i,k(\ell))P^{K(\ell)}_\ell, \tag{C.4}$$

and finally the resultant model after the rearrangement can be represented by the following formula, which is equal to Eq. (13).

$$\begin{aligned} M_n &= \prod_{j=0}^{n} \otimes^i_j Z^i_j \\ &= \otimes_0 P_0 \prod_{j=1}^{n} \otimes^i_j \left(P^i_j \prod_{\ell=0}^{j-1} \underline{\otimes}(j,\ell)\alpha(j,\ell)\beta(i,k(\ell))P^{k(\ell)}_\ell \right). \end{aligned} \tag{C.5}$$

References

[1] Lee K. Principles of CAD/CAM/CAE systems. Addison Wesley Longman, Inc.; 1999.
[2] Shah JJ, Mäntylä M. Parametric and feature-based CAD/CAM. John Wiley & Sons, Inc.; 1995.
[3] Sun G. A digital mock-up visualization system capable of processing giga-scale CAD models. Computer-Aided Design 2007;39(2):133–41.
[4] Cignoni P, Montani C, Scopigno R. A comparison of mesh simplification algorithms. Computers & Graphics 1998;22(1):37–54.
[5] Lee SH. Feature-based multiresolution modeling of solids. ACM Transaction on Graphics 2005;24(4):1417–41.
[6] Cera CD, Kim T, Han J, Regli WC. Role-based viewing envelopes for information protection in collaborative modeling. Computer-Aided Design 2004;36(9):873–86.
[7] Koo S, Lee K. Wrap-around operation to make multi-resolution model of part and assembly. Computers & Graphics 2002;26(5):687–700.
[8] Lee JY, Lee J-H, Kim H, Kim HS. A cellular topology-based approach to generating progressive solid models from feature-centric models. Computer-Aided Design 2004;36(3):217–29.
[9] Li WD, Lu WF, Fuh JYH, Wong YS. Collaborative computer-aided design—research and development status. Computer-Aided Design 2005;37(9):931–40.
[10] Chu C-H, Chan Y-H, Wu PH. 3D streaming based on multi-LOD models for networked collaborative design. Computers in Industry 2008;59(9):863–72.
[11] Armstrong CG. Modelling requirements for finite-element analysis. Computer-Aided Design 1994;26(7):573–8.
[12] Thakur A, Banerjee AG, Gupta SK. A survey of CAD model simplification techniques for physics-based simulation applications. Computer-Aided Design 2009;41(2):65–80.
[13] Lee SH. A CAD-CAE integration approach using feature-based multiresolution and multi-abstraction modelling techniques. Computer-Aided Design 2005;37(9):941–55.
[14] Li B, Liu J. Detail feature recognition and decomposition in solid model. Computer-Aided Design 2002;34(5):405–14.
[15] Lee SH, Lee K. Feature-based multiresolution techniques for product design. Journal of Zhejiang University Science A 2006;7(9):1535–43.
[16] Lee SH, Lee K. Partial entity structure: a compact boundary representation for non-manifold geometric modeling. ASME Journal of Computing & Information Science in Engineering 2001;1(4):356–65.
[17] Bidarra R, de Kraker KJ, Bronsvoort WF. Representation and management of feature information in a cellular model. Computer-Aided Design 1998;30(4):301–13.
[18] Choi DH, Kim TW, Lee K. Multiresolutional representation of *b*-rep model using feature conversion. Transactions of the Society of CAD/CAM Engineers 2002;7(2):121–30 [In Korean].
[19] CATIA V.5: Help, Dassault Systemes. 2008.
[20] Unigraphics NX3: Modeling Help. UGS Corp. 2004.
[21] SolidWorks 2008: Help. SolidWorks Corp. 2008.
[22] Belaziz M, Bouras A, Brun JM. Morphological analysis for product design. Computer-Aided Design 2000;21(5–6):377–88.
[23] Lee SH, Lee K-S, Park S. Multiresolution representation of solid models using the selective Boolean operations. In: Proceedings of the 2002 spring conference of Korean society of precision engineering. 2002. p. 833–5 [In Korean].
[24] Lee SH, Lee K-Y, Woo Y, Lee K-S. Feature-based multi-resolution modeling of solids using history-based Boolean operations—Part I: theory of history-based Boolean operations. Journal of Mechanical Science and Technology 2005;19(2):549–57.
[25] Kim SC, Lee K, Hong T, Kim M, Jung M, Song Y. An integrated approach to realize multiresolution of *B*-rep model. In: Proceedings of the 2005 ACM symposium on solid and physical modeling. Cambridge, MA, USA: MIT; 2005. p. 153–62.
[26] Lee SH, Lee K, Kim SC. History-based selective Boolean operations for feature-based multi-resolution modeling. In: Proceedings of international conference on computational science and its applications, Part I. 2006. pp. 459–68.
[27] Woodwark JR, Quinlan KM. Reducing the effect of complexity on volume model evaluation. Computer-Aided Design 1982;14(2):89–95.
[28] Tilove RB. A null-object detection algorithm for constructive solid geometry. Communications of the ACM 1984;27(7):684–94.
[29] Rossignac JR, Voelcker HB. Active zones in CSG for accelerating boundary evaluation, redundancy elimination, interference detection, and shading algorithms. ACM Transactions on Graphics 1988;8(1):51–87.
[30] Raghothama S, Shapiro V. Consistent updates in dual representation systems. Computer-Aided Design 2000;32(8–9):463–77.
[31] Masuda H. Topological operators and Boolean operations for complex-based nonmanifold geometric models. Computer-Aided Design 1992;25(2):119–29.
[32] Mäntylä M. An introduction to solid modeling. Computer Science Press; 1988.
[33] Crocker GA, Reinke WF. An editable non-manifold boundary representation. IEEE Computer Graphics & Applications 1991;11(2):39–51.
[34] Sheen D-P, Son T-G, Myung D-K, Ryu C, Lee SH, Lee K, Yeo TJ. Transformation of a thin-walled solid model into a surface model via solid deflation. Computer-Aided Design 2010;42(8):720–30.
[35] Bidarra R, Madeira J, Neels WJ, Bronsvoort WF. Efficiency of boundary evaluation for a cellular model. Computer-Aided Design 2005;37(12):1266–84.
[36] Anderson B, Andnettles D. Product data representation and exchange: integrated application resource: construction history features. ISO/CD 10303-111. 2003.
[37] Lee SH. Offsetting operations on non-manifold topological models. Computer-Aided Design 2009;41(11):830–46.
[38] Lin S-YT, Lin Y-F. Set theory: an intuitive approach. Boston: Houghton Mifflin; 1974.