

# Volumetric modification of solid CAD models independent of design features

Yoonhwan Woo<sup>a</sup>, Sang Hun Lee<sup>b,\*</sup>

<sup>a</sup> Department of Mechanical Systems Engineering, Hansung University 389 Samsun-dong 3ga, Seongbuk-gu Seoul 136-792, Republic of Korea

<sup>b</sup> Graduate School of Automotive Engineering, Kookmin University, School of Mechanical and Automotive Engineering, 861-1 Jeongneung-dong, Seongbuk-gu, Seoul 136-702, Republic of Korea

Received 9 January 2006; received in revised form 28 March 2006; accepted 17 April 2006

Available online 9 June 2006

## Abstract

When feature information is lost during a data transfer, the advantage of feature-based modeling becomes invalid. After that, even a small modification for a feature of the model becomes purely dependent on geometric and topological manipulations. The existing methods for the local modifications such as moving and tapering faces basically make use of ‘tweaking’, which is the technique to replace the surface of a face with different one. However, tweaking is limited to the modifications that do not involve topological changes in the model. Due to this limitation, tweaking does not work properly when local modifications involve topological changes. In this paper, a new volumetric method for the local modification for 3D CAD models is presented. This method can handle local modification involving topological changes by first separating the volume of feature from the model using the volume decomposition, and then uniting the modified feature with the model.

© 2006 Elsevier Ltd. All rights reserved.

*Keywords:* Local modification; Selective volume decomposition; Solid modeling

## 1. Introduction

With the advancements of the Internet and CAD data translation techniques, more CAD models are transferred from a CAD system to another through the network, and ‘interoperability’ is getting a common word in the CAD industry. When CAD models are transferred to an incompatible CAD system, the data has to be translated so that they can be properly restored for the system. The interoperability helps heighten the status of the concurrent engineering and reduce the time for the product development cycle. However, the interoperability is not complete in the sense that only topology and geometry of the models are translated. The true interoperability should have the ability to manipulate not only geometry and topology, but also

feature information on top of them. But currently the precious feature information is being lost during the CAD data translation. Neutral formats for CAD data translation such as STEP presently provide no ways to represent the feature information [1]. Although the ISO committee associated with STEP decided to create a second edition that could convey more comprehensive design information such as construction history features, at present, there is no such CAD model translators that can handle the construction history features [1].

When feature information is lost during a data transfer, the advantage of feature-based modeling is not valid any longer. Now modifications for the models are purely dependent on geometric and topological manipulations. The existing methods for the local modifications such as moving and tapering faces basically make use of ‘tweaking’, which is the technique to replace the surface of a face with different one [2]. When the surface of a face is to be changed, the geometry of the related entities such as edges

\* Corresponding author. Tel.: +82 2 910 4835; fax: +82 2 916 0701.  
E-mail addresses: [ywoo@hansung.ac.kr](mailto:ywoo@hansung.ac.kr) (Y. Woo), [shlee@kookmin.ac.kr](mailto:shlee@kookmin.ac.kr) (S.H. Lee).

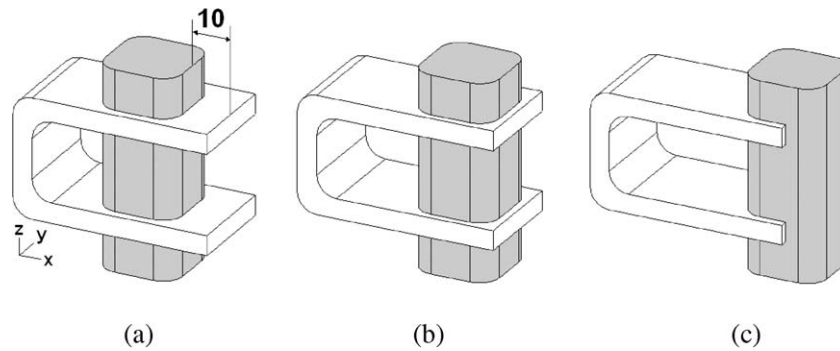


Fig. 1. Move by tweaking: (a) a protrusive feature on a translated model; (b) move the boss along  $x$ -axis by 9 units—no topological changes necessary: tweaking succeeds; (c) move the boss along  $x$ -axis direction by 20 units—topological changes necessary: tweaking fails.

and vertices also need to be changed. The tweaking operation calculates the curves of the new edges by intersecting the face's new surface with the surfaces of adjacent faces, and the positions of the new vertices by intersecting each new curve with its neighbors. As long as the topology of the entities remains the same, it is straightforward to calculate their new geometry. But the cases are not always so. In many cases, local modifications cause topological changes, and they fail. Since there is no generalized way to handle the topological changes, only the limited cases are handled by incorporating some heuristics and rules.

In Fig. 1(a), for example, suppose a user needs to move the faces of the protrusive feature on the translated model by 9 in the  $x$ -axis direction. This modification does not involve any topological change of the faces, and the geometry of new edges and vertices can be simply calculated with the adjacent faces of new surfaces. So the move by tweaking succeeds in this case as shown in Fig. 1(b). However, what if the user wants to move the faces of the feature by 20 in the  $x$ -axis direction as shown in Fig. 1(c)? In this case, the modification fails.<sup>1</sup> The feature should move beyond the side face of the model, so the topological changes need to be handled accordingly. But since tweaking only considers the faces to be changed and the faces adjacent to them, the information for the topological manipulations is not acquired. In order to get the model by tweaking as shown in Fig. 1(c), it is necessary to consider all the faces of the model, and check if which faces intersect with which faces, and then tidy up the relationship to correctly generate new edges and curves. However, this approach would be very complicated and implausible, and the local modification would not be 'local' any more. It is becoming more like a Boolean operation requiring a 'global' reasoning for the intersections of all the faces of a model.

Now suppose we were able to separate the volume of the feature from the model as shown in Fig. 2(b). In this case, it will be straightforward to modify the separated feature without concerning about the topological changes. The

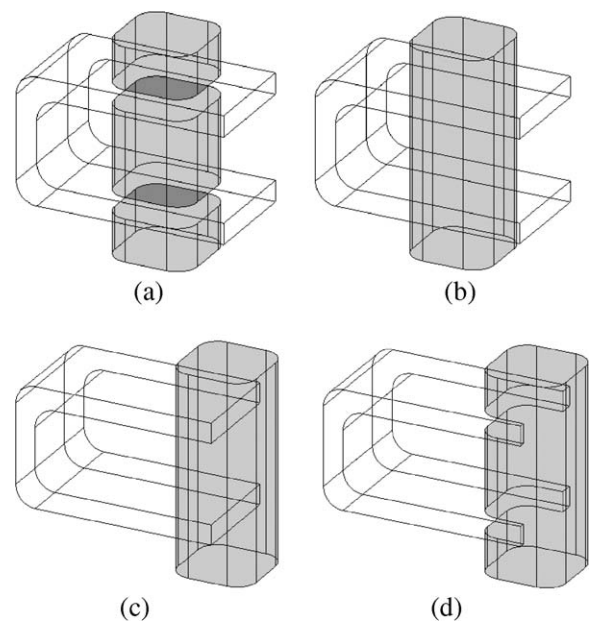


Fig. 2. Move by volumetric approach: (a) wire-frame view of the model in Fig. 1(a); (b) separation of the volume of feature by volume decomposition; (c) move the feature volume by 20; (d) union of the feature and the model.

move operation can be simply done by first translating the separated feature by 20 as shown in Fig. 2(c), and then uniting it with the remaining volume by a Boolean operation. The resultant model is shown in Fig. 2(d). By doing this, the local modification involving topological changes can be achieved without considering all the faces of the model to calculate new edges.

In this paper, we propose a new volumetric approach to local modifications for data exchanged 3D solid models. The overview of this approach is illustrated in Fig. 3. This approach comprises of three steps: separating feature by selective volume decomposition, modifying feature, and uniting the modified feature. It is noted that the method presented in this paper is limited to modifications of protrusive features. In addition, the faces with concave edges should be analytic surfaces so they can be extended uniquely. In the following sections, each step is explained in detail.

<sup>1</sup> With the ACIS 3D modeling kernel, the operation results in an invalid model.

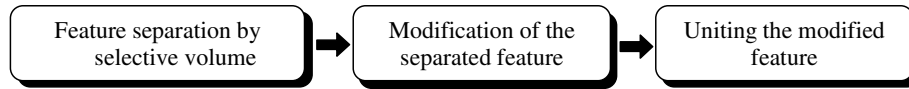


Fig. 3. Overview of the volumetric approach to local modification.

## 2. Feature separation by selective volume decomposition

There were several research efforts to decompose a solid model into simpler volumes [3–6]. Most of them are found in the area of machining feature recognition. These methods were used to decompose the delta volume—the volumetric difference between the part and the stock—of machined parts into the volumes of simple machining features. These volume decomposition methods have been used in other engineering applications [7–9]. Among them, maximal volume decomposition (MVD) [3,4] is an existing method to decompose a solid model into the sub-volumes called maximal volumes. Volume  $V$  is a maximal volume of solid model  $S$  if  $V$  satisfies all the following conditions:

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

Separation of feature of interest—the key process of the method to be presented in this paper—is primarily dependent on this maximal volume decomposition [3,4]. However, this method involves several new techniques to accomplish to separation of features of interest from data exchanged CAD models. These are:

- Selective volume decomposition

Although it is possible to separate features of interest by using the ordinary maximal volume decomposition, it may become computationally very heavy as it extends all the faces and intersect them to create a cellular model. The method presented in this paper is capable of decomposing a solid model into sub-volumes by extending and intersecting only the faces selected by users. This capability significantly reduces the computational load and provides a more efficient way to handle the feature modifications.

- Application of decomposition method to model modification

Although the volume decomposition method is a general method to decompose a complex solid model into simpler volumes, it has been mainly used in the area of machining feature recognition. For machining feature recognition, a delta volume—difference between a part and the stock—is decomposed into machining features that are all *subtractive* features. The method proposed in this paper applies the decomposition method to the model modifications. It selectively decomposes the model itself to modify protrusive features that are *additive* features.

The first step of MVD is to extend all the faces having concave edges and intersect the extended faces with the model. This gives a non-manifold model that has multiple 3D regions partitioned by the extended faces. These regions are formed into cells each of which contains topological entities that enclose a void. Each cell points to a CSHELL that is a topological entity bounding the 3D region of the cell. A CSHELL contains a list of CFACES. CFACE is a face of cell that points to a face in the model with a forward or reversed sense. CFACES can be classified into the following three types based on their underlying surfaces and senses: R-CFACE, E-CFACE, and C-CFACE. Here, R-, E-, and C- stand for *Real*, *Extended*, and *Complement*, respectively (see Fig. 4).

The faces of original solid model are denoted by  $Fo_i$ , where  $i = 1, \dots, N$ , and a CFACE  $Fc$  ( $N$  is the number of faces of the original model). It is defined that  $Fc \uparrow Fo_k = 1$  if the surface geometry of  $Fo_k$  and  $Fc$  is identical, and they use the same side of the surface. Similarly,  $Fc \uparrow Fo_k = -1$  if the surface geometry of  $Fo_k$  and  $Fc$  is identical, and they use the opposite sides of the surface.

- A CFACE  $Fc$  is R-CFACE if the underlying face of  $Fc$  is a part of a face of the original model.

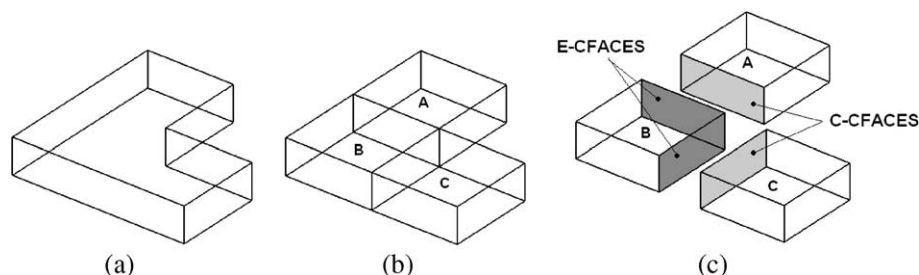


Fig. 4. Classification of CFACES: (a) solid model; (b) cells created by face extension; (c) CFACES (all the unmarked CFACES are R-CFACES).

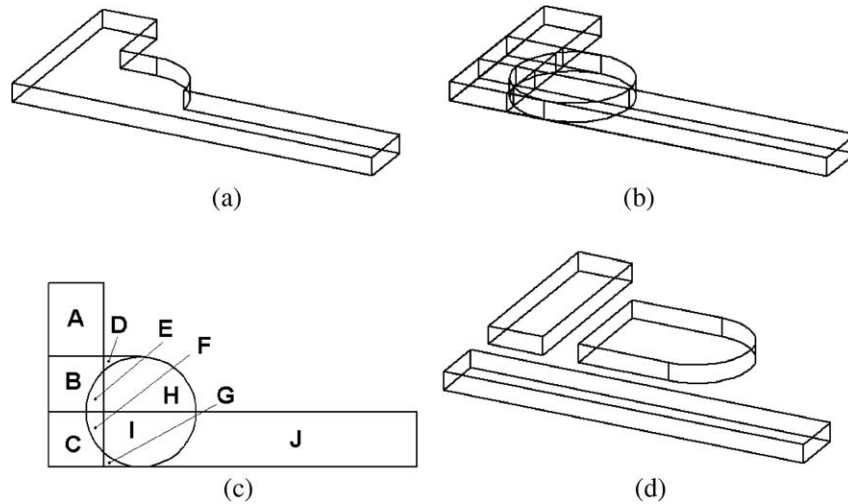


Fig. 5. An example of maximal volume decomposition: (a) a part; (b) the cellular model of the part; (c) top view of the cellular model; (d) maximal volumes of the part.

- A CFACE  $F_c$  is E-CFACE if it satisfies the following conditions:
  - $\{x|F_c \uparrow F_{o_x} = 1\} \neq \emptyset$ .
  - $\{y|F_c \uparrow F_{o_y} = -1\} = \emptyset$ .
  - The underlying face of  $F_c$  is not a part of a face of the original model.
- A CFACE  $F_c$  is C-CFACE if it satisfies the following conditions:
  - $\{x|F_c \uparrow F_{o_x} = 1\} = \emptyset$ .
  - $\{y|F_c \uparrow F_{o_y} = -1\} \neq \emptyset$ .
  - The underlying face of  $F_c$  is not a part of a face of the original model.

There can be a case where a CFACE satisfies the conditions for not only E-CFACE but also C-CFACE. This happens when a model has two faces that share the same surface geometry, but they use the different sides of the surface. This kind of CFACE is classified as S-CFACE (S-stands for *special*). The occurrence of S-CFACE is relatively very low.

When a solid model is decomposed into cells by the method explained above, there always exist the cells that have only R-CFACES and C-CFACES. These cells are called *seed cells*. For identification of seed cells, a S-CFACE is considered as C-CFACE. For instance, the cellular model of the part in Fig. 5(a) is generated by extending the faces and intersecting them with the model as shown in Fig. 5(b). Fig. 5(c) shows the top view of the cellular model. In this case, the cellular model has 10 cells of which A, D, G and J are the seed cells.<sup>2</sup>

Once the cellular model is generated, the next step is to collect certain cells that constitute a maximal volume. It is noted that a maximal volume is a combination of cells, and it always has at least one seed cell. Therefore, a maximal volume can be generated by adding cells to a seed cell such that their union satisfies the conditions for maximal volume. Fig. 6 shows the scheme of the method that collects cells for a maximal volume with a given seed cell. In the scheme,  $L_{all}$  denotes a list that will include all the cells constituting a maximal volume at the end.  $L_{iter}$  denotes a list that will include only the cells satisfying the collecting condition that are adjacent to an input cell.

The algorithm in Fig. 6 is explained with the example model in Fig. 5. For the seed cell A, it is first added to  $L_{all}$ . Since the cell B is in contact with the C-CFACE of the cell A, it is added to  $L_{all}$  and  $L_{iter}$ . There are no other cells to be added for the seed cell A, so the process goes to the step (12) where the cell B in  $L_{iter}$  is given to the step (2) as input. For the cell B, the cells C and E are added to  $L_{all}$  and  $L_{iter}$  since they are in contact with C-CFACES. However, the cell D in contact with the E-CFACE of the cell B is not added, because it does not satisfy the condition in the step (7). Again, the process goes to the step (12), where the cells C and E in  $L_{iter}$  are given to the step (2) as input.

Now for the cell C, it has one C-CFACE in contact with the cell F, so the cell F is added to  $L_{all}$  and  $L_{iter}$ . The cell G is in contact with an E-CFACE of the cell C, but it is not added due to the condition in the step (7). There are no other cells to be added for the cell C. For the cell E, it has the E-CFACE in contact with the cell H, but the cell H is not added due to the condition in the step (7). The cell F is in contact with the C-CFACE of the cell C, but it is not added either since it has been already added to the lists (condition in the step (8)). Now the cell F is the only cell in  $L_{iter}$ , which is given to the step (2) as input. There are no cells to be added for the cell F, so the process terminates. Starting with the seed cell A, the cell B, C, E, and

<sup>2</sup> A has 5 R-CFACES and 1 C-CFACE; B has 3 R-CFACES, 2 E-CFACES, and 2 C-CFACES; C has 4 R-CFACES, 2 E-CFACES, and 1 C-CFACE; D has 3 R-CFACES and 1 C-CFACE; E has 2 R-CFACES, 2 E-CFACES, and 1 C-CFACE; F has 2 R-CFACES, 2 E-CFACES, and 1 C-CFACE; G has 3 R-CFACES and 1 C-CFACE; H has 3 R-CFACES, 1 E-CFACE, and 2 C-CFACES; I has 2 R-CFACES, 3 E-CFACES, and 1 C-CFACE; J has 5 R-CFACES and 1 C-CFACE.

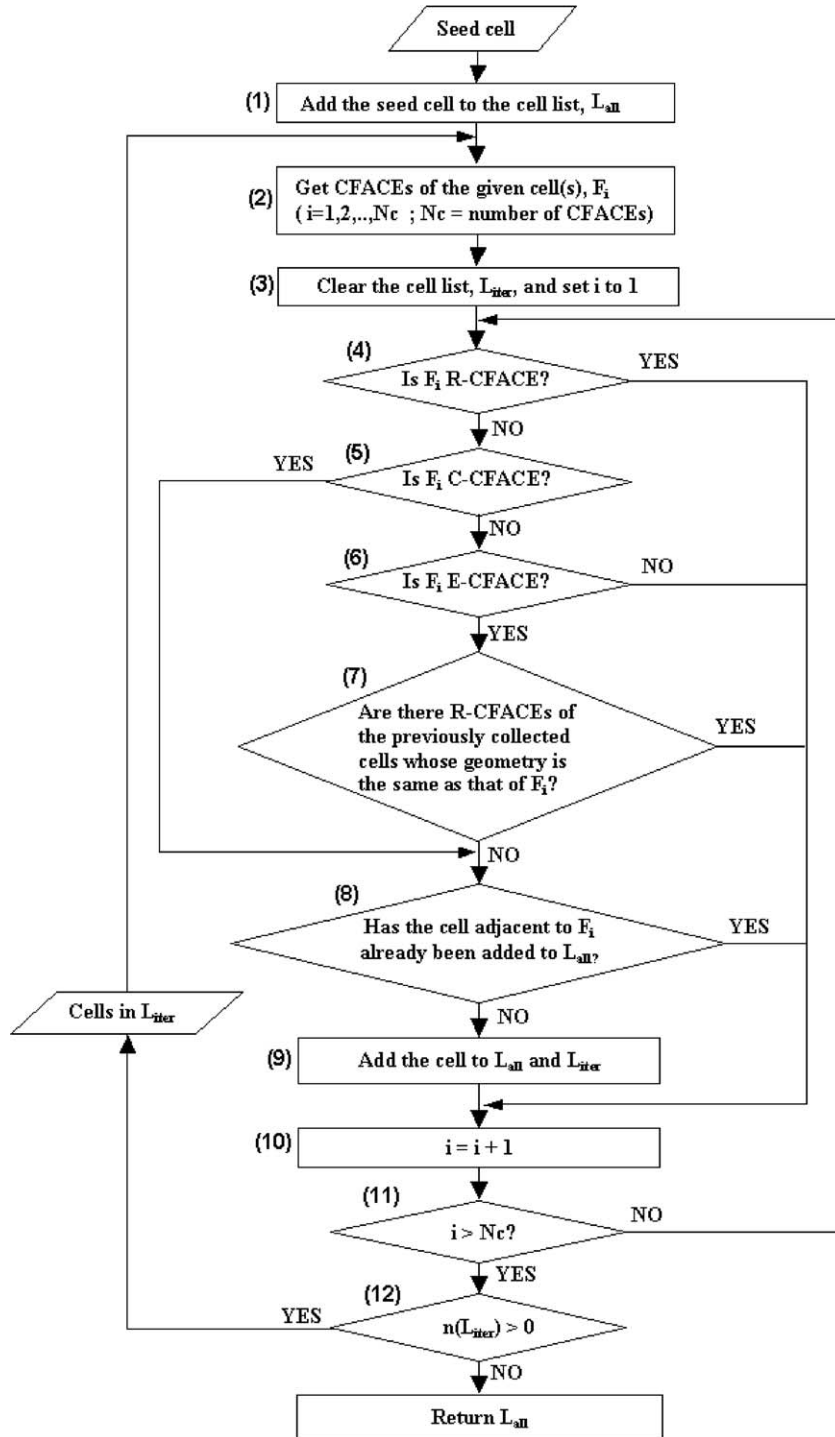


Fig. 6. Method for collecting cells constituting a maximal volume from a seed cell.

F are collected. The cells A-B-C-E-F are composed into a maximal volume in Fig. 5(d).

Likewise, for the seed cell D, the cell B, C, E, F, G, H, and I are collected. The seed cell G is not used since it is contained in a collection. For the seed cell J, the cell C, F, G, and I are collected. By composing the cells in each collection, three maximal volumes are generated for the model as shown in Fig. 5(d). How to compose the cells can be found in [10].

For the purpose of separation of a specific feature, however, it is not necessary to extend all the faces to create cells. Only a set of faces necessary for separating the feature from the model are used. Those faces will be the faces of the feature that user selects and their adjacent faces through concave edges. By extending only these faces and intersecting them with the model, the cellular model that is sufficient for the feature separation can be generated. The subsequent operations will be the same as the regular

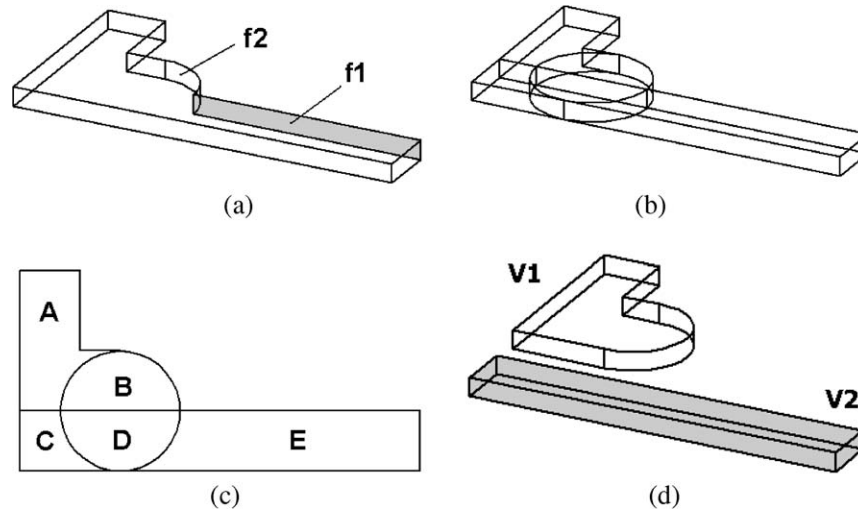


Fig. 7. Selective volume decomposition with the face f1: (a) part; (b) cellular model with selective face extension; (c) top view of the cellular model; (d) sub-volumes generated by selective volume decomposition.

MVD except that the concave edges on non-selected faces are treated as “not-concave” in the cell collection.

Suppose a user picks the face f1 in Fig. 7(a). The face f2 is adjacent to f1 through the concave edge. Therefore, the face f1 and f2 are extended and intersected to form the cellular model shown in Fig. 7(b). In this case, the cellular model has only five cells of which the cell A and E are the seed cells. Based on the algorithm shown in Fig. 6, the cells B, C, and D are added to the seed cell A, and are composed into the sub-volume V1 in Fig. 7(d). Similarly, the cells C and D are added to the seed cell E, and are composed into the sub-volume V2. With this selective volume decomposition, only two sub-volumes are generated. It is noted that a sub-volume generated by the selective volume decomposition is not necessarily a maximal

volume, since not all the faces having concave edges are extended.

Once sub-volumes are generated, we need to identify which sub-volume is the volume of the separated feature. The volume of the separated feature can be identified by checking if a sub-volume contains a face picked by the user. If so, it is the volume of the separated feature. The selective volume decomposition may produce more than two sub-volumes. In this case, the sub-volumes that are not the volume of feature are united together, so we have only two volumes: the volume of the separated feature and the volume of the model with the feature separated. From now on, the volume of the separated feature is called the *feature volume*, and the volume of the model with the feature separated the *base volume*.

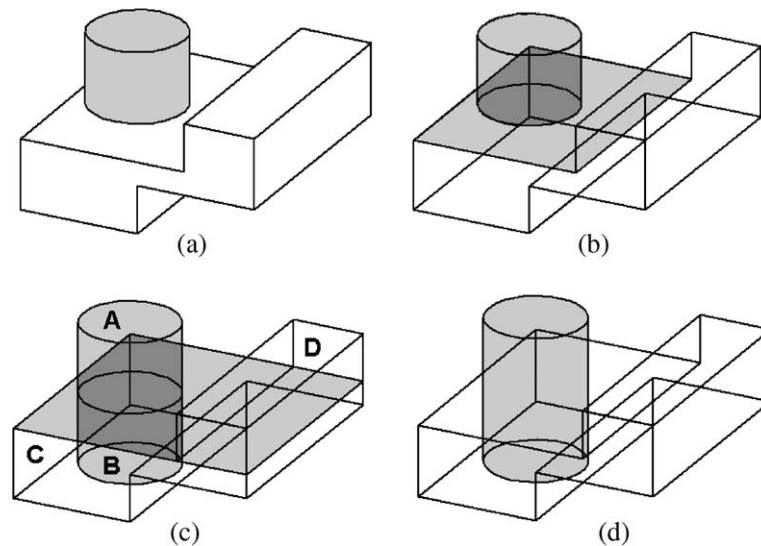


Fig. 8. Feature separation by selective volume decomposition: (a) a model with the feature to be separated; (b) gray faces are to be extended and intersected; (c) cells; (d) feature separated.

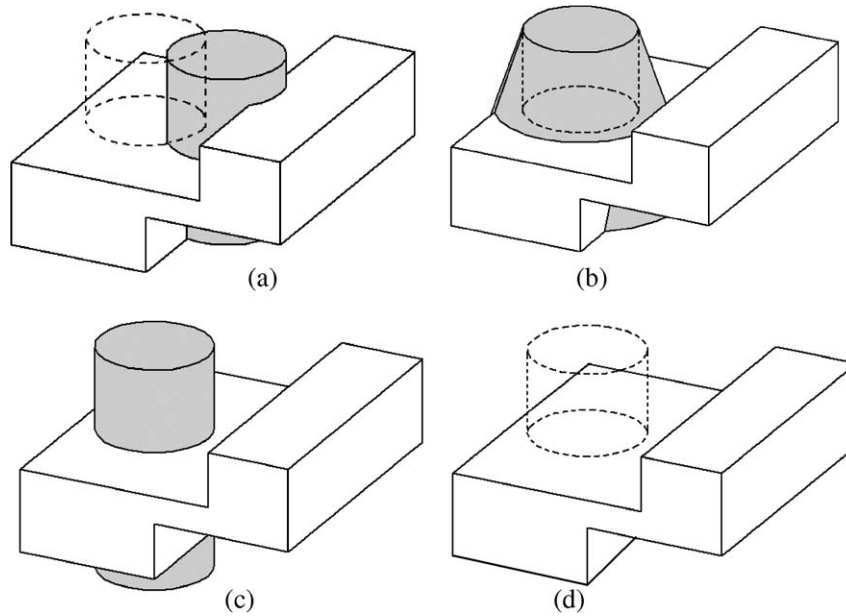


Fig. 9. Some possible modifications for the separated feature: (a) move by translation; (b) tapering by tweaking; (c) extension by scaling; (d) deletion.

Consider the model in Fig. 8(a) for instance. In order to separate the cylindrical boss, a user picks the faces denoted by gray color. The planar face adjacent to the picked cylindrical face through concave edge is also selected for the decomposition as shown in Fig. 8(b). Next, the selected faces are extended, and they are intersected with the model to generate the cellular model shown in Fig. 8(c). This cellular model has four cells of which the cell A and the cell D are the seed cells. For the seed cell A, the cell B is added, and they are composed into the feature volume. For the seed cell D, the cell C and B are added, and they are composed into the base volume in this case. Fig. 8(d) shows the feature volume and the base volume for the model.

**3. Modifications for separated feature**

Now that the feature volume is separated from the model, a modification for the feature is only limited to the feature itself. It is free to modify the feature without concerning about the topological changes caused by possible interactions between the feature volume and the base volume. This allows us to do various modeling operations. Some of them are shown in Fig. 9.

In Fig. 9(a), the feature is separated and then moved by a simple translation operation. In Fig. 9(b), the feature is separated and tapered by a tweaking operation. In Fig. 9(c), the feature is separated and extended by a scaling operation. In Fig. 9(d), the feature is separated and removed from the model.

**4. Uniting feature volume**

Once a modification for the feature volume is completed, the next step is to uniting it with the base volume. For the example in Fig. 9(a), it can be done by a simple Boolean unite operation. However, it should be noted that there may be alternative ways of uniting the feature volume with the base volume depending on the user’s intention. For example, the user may want to have the feature modified either as shown in CASE I or as shown in CASE II in Fig. 10 based on the user’s modification purpose. If CASE I is the choice of the user, it can be simply done by uniting the feature volume with the base volume. However, if CASE II is the choice of the user, it is not as straightforward. In this case, only some portions of the feature volume should be united with the base volume. Therefore, it

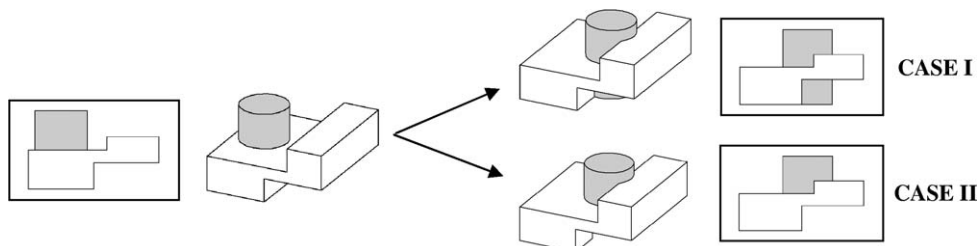


Fig. 10. Alternative results based on user’s intention.

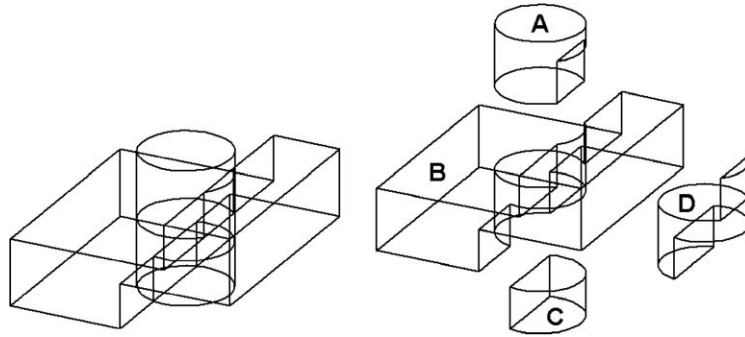


Fig. 11. Cells created by non-regularly uniting the two volumes.

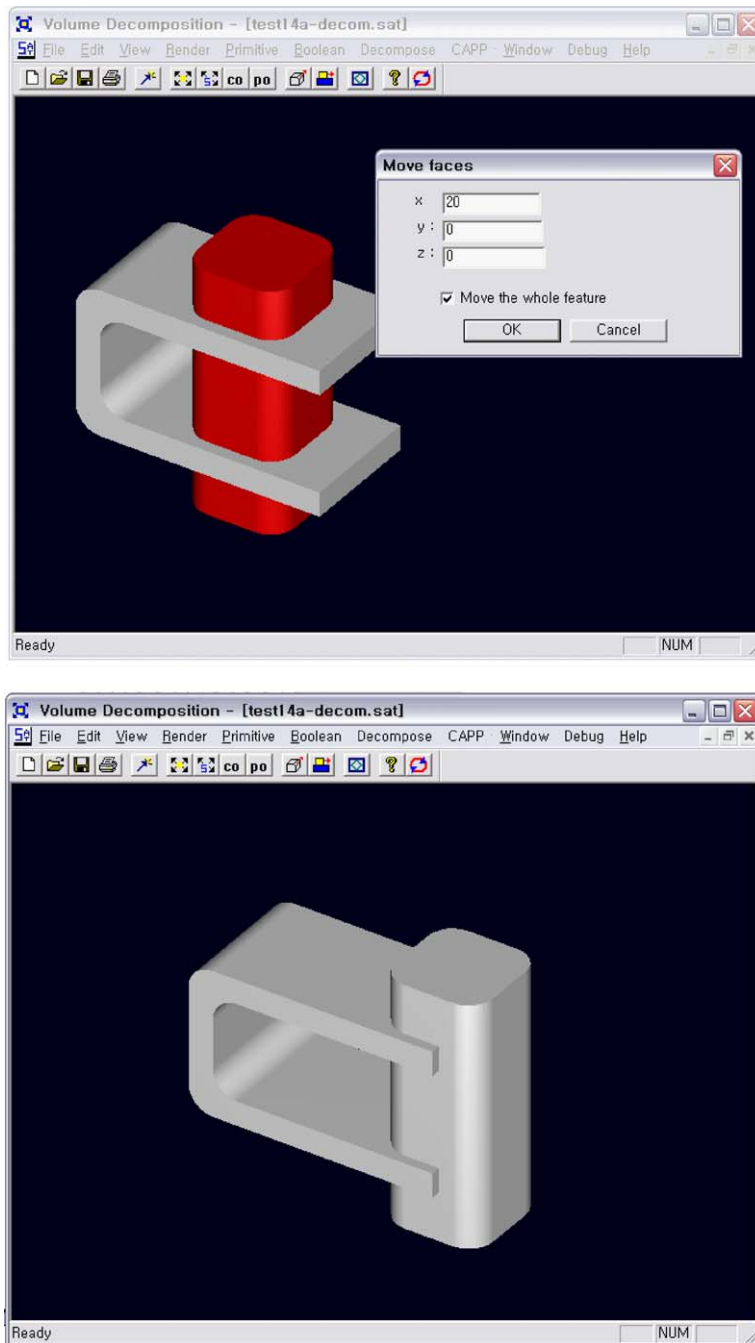


Fig. 12. Local modification by feature separation: move operation.



is necessary to provide the user with the function for selecting the desired portions of the feature volume.

For this selecting function, it is needed to partition the feature volume using the base volume. The partition of the feature volume can be achieved by a non-regularized Boolean unite operation. The non-regularized Boolean unite operation for the feature volume and the base volume gives a non-manifold model, from which the cells enclosing the portions partitioned by the faces can be generated. For the model in Fig. 10, four cells are generated by non-regularly uniting the feature volume and the base volume as shown in Fig. 11. From these four cells, since only A and C are the cells that are originated from the feature volume, and are not contained by the base volume, these two cells

are prompted for the user's selection. By selecting only A, we can get the model shown in the case II in Fig. 10. If there is only one cell that is originated from the feature volume and is not contained by the base volume, there is only a single choice.

## 5. Implementation and examples

The method presented in this paper has been implemented using C++ on top of ACIS running on a Windows PC. For the model shown in Fig. 12, a user selects the faces of protrusion feature to be modified. When the selection is done, a pop-up window prompts the user to input the distances to move in  $x$ -,  $y$ -, and  $z$ -directions. The boss features

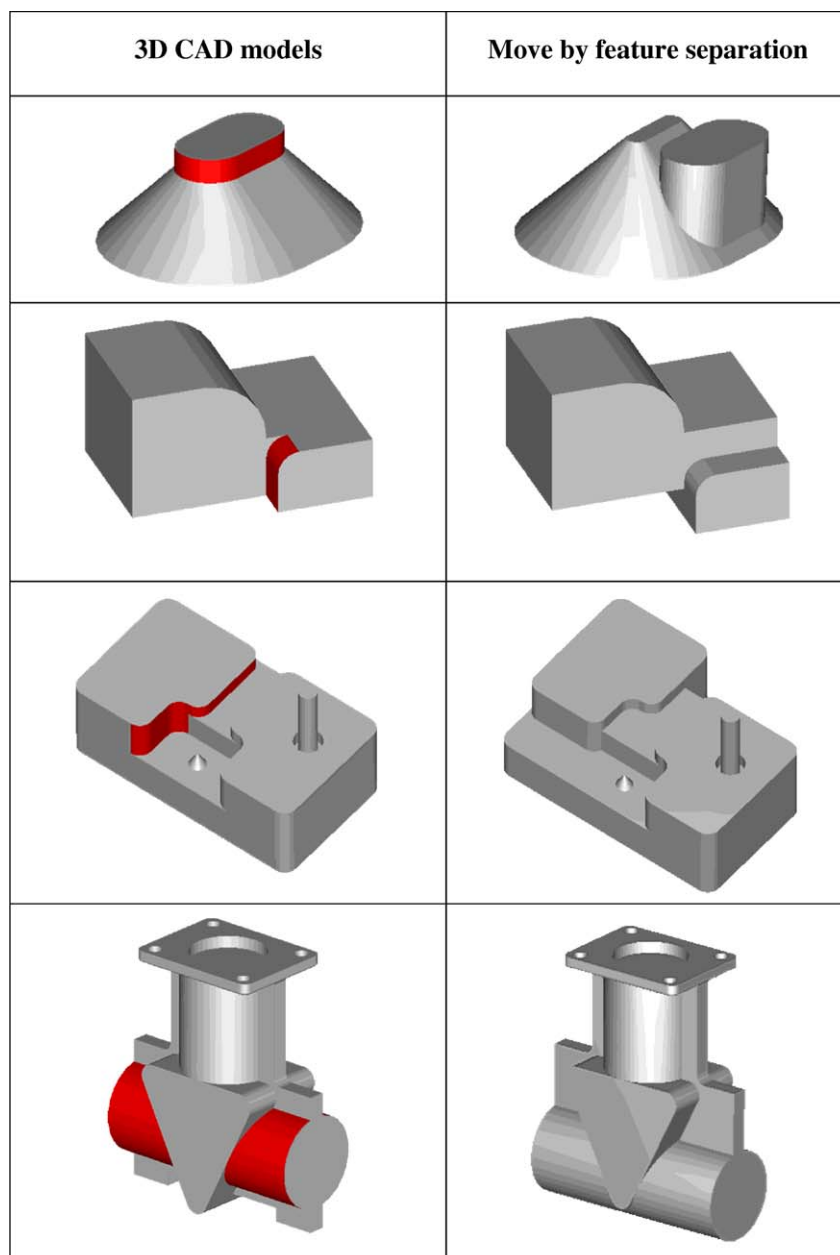


Fig. 13. Examples of local modifications of data exchanged parts by the feature separation with the selective volume decomposition. The faces in red are picked by user.

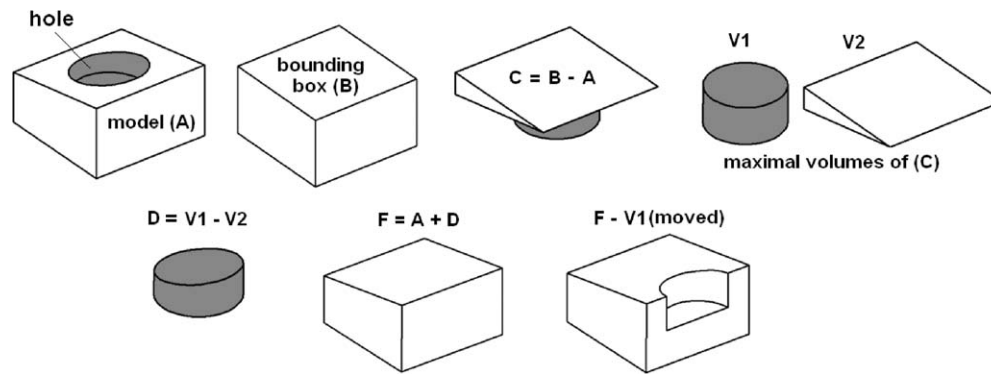


Fig. 14. Basic concept of modification of depression features.

is successfully moved by 20 along the  $x$ -axis direction. Fig. 13 shows more examples of local modifications for data exchanged CAD models. These modifications are not possible with the conventional tweaking operations.

## 6. Conclusion

In this paper, a new method for local modification for solid models has been presented. It is summarized as follows:

- The method presented in this paper, which is based on the selective volume decomposition, can separate the feature of interest from the model, so that modifications for the feature can be accomplished without concerning about the topological changes. The method can complement the shortcomings of the current tweaking method. Since this method is independent of feature-based models, it allows the modifications for the static CAD model whose feature information has been lost during data transfers.
- The selective volume decomposition is a general method for solid modeling. Therefore, it can be used for other modeling operations such as design feature restoration and model simplification.
- Presently, the method only can handle the protrusive features that have boundaries of concave edges. If a boundary of concave edges of protrusive feature gets blended into other features such as fillets, preliminary operations to remove blending features may be necessary for practical use. The existing methods for removing blending features [11,12] can be used for this purpose.
- Modification of depression features currently has not been addressed. Further research needs to be made to address this limitation. The basic concept to accomplish the modification of depression features would be similar to that of protrusive features. However, since a depression feature cannot be represent as a volume that is a subset of solid model, it would be necessary to introduce a new solid from which the volume equivalent of the depression feature can be extracted. Fig. 14 shows this basic concept with a model with simple hole. It should be noted that this idea is primitive.

## Acknowledgements

For the first author, this research was financially supported by Hansung University in the year of 2006. For the second author, this work is financially supported by the Ministry of Education and Human Resources Development(MOE), the Ministry of Commerce, Industry and Energy(MOCIE) and the Ministry of Labor(MOLAB) through the fostering project of the Lab of Excellency.

## References

- [1] Pratt MJ et al. Towards standardized exchange of parameterized feature-based CAD models. *Comput Aided Des* 2005;37(12):1251–1265.
- [2] ACIS® 3D modeling kernel online help, Spatial Corp, Available from: <http://www.spatial.com>.
- [3] Sakuri H, Dave P. Volume decomposition and feature recognition, Part II: curved objects. *Comput Aided Des* 1996;28(6/7):519–537.
- [4] Woo Y, Sakurai H. Recognition of maximal features by volume decomposition. *Comput Aided Des* 2002;34(3):195–207.
- [5] Shah J, Shen Y, Shirur A. Determination of machining volumes from extensible sets of design features. In: Shah J, Mantyla M, Nau D, editors. *Advances in feature based manufacturing*. Amsterdam: Elsevier; 1994. p. 129–57.
- [6] Wang E, Kim Y. Form feature recognition using convex decomposition: results presented at the 1997 ASME CIE feature panel session. *Comput Aided Des* 1998;30(13):983–9.
- [7] Lu Y, Gadh R, Tautges T. Feature based hex meshing methodology: feature recognition and volume decomposition. *Comput Aided Des* 2001;33(3):221–32.
- [8] Kailash S, Zhang Y, Fuh J. A volume decomposition approach to machining feature extraction of casting and forging component. *Comput Aided Des* 2001;33(8):605–17.
- [9] Litte G, Clark D, Corney J, Tuttle J. Delta-volume decomposition for multi-sided components. *Comput Aided Des* 1998;30(9):695–705.
- [10] Woo Y. Fast cell-based decomposition and applications to solid modeling. *Comput Aided Des* 2003;35(11):969–77.
- [11] Venkataraman S, Sohoni M. Blend recognition algorithm and applications. In: *Proceedings of ACM solid modeling conference*, Ann Arbor, MI, USA; 2001. p. 99–108.
- [12] Zhu H, Meng CH. B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. *Comput Aided Des* 2002;34(2):109–23.