Springer

KSME

Journal of Mechanical Science and Technology 28 (5) (2014) 1759~1764
www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-014-0322-0

# Protrusion recognition from solid model using orthogonal bounding factor†

Yoonhwan Woo* and Sang-Hyun Kim
*Department of Mechanical Systems Engineering, Hansung University, Seoul, 136-792, Korea*

(Manuscript Received June 17, 2013; Revised November 22, 2013; Accepted January 10, 2014)

**Abstract**

Feature recognition is important for describing shapes in many applications taking advantage of solid modeling. Graph-based feature recognition methods search from solid models the unique patterns of features that are represented as a graph. A typical example of such patterns is a loop of concave edges. When the loop is an inner loop on a single face, it is a strong hint of the existence of protrusion feature and recognition of protrusion faces is straightforward. However, when a protrusion feature lies on multiple faces, it is bounded by a loop of concave edges that are not on a single face. Consequently, the rule of inner loop is no more available and recognition of protrusion faces becomes unclear. To address this problem, a new quantitative measure, orthogonal bounding factor (OBF), is introduced. OBF is defined as the sum of cross products of two consecutive vectors normal to a set of faces, and it physically represents the possibility of being a protrusion in a solid model. The formal definition of orthogonal bounding factor is established and a method to recognize protrusion features using OBF is presented. Examples are also shown to demonstrate the effectiveness of the method for feature recognition.

*Keywords*: Feature recognition; Graph matching; Orthogonal bounding factor; Protrusion feature; Quantitative measure

## 1. Introduction

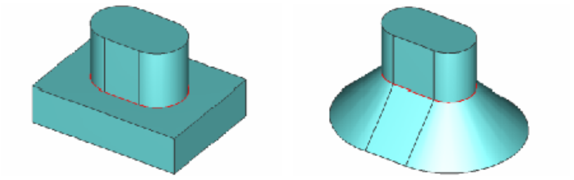
Feature recognition is important for describing shapes in many applications taking advantage of solid modeling. Accordingly, there have been extensive research efforts in the area, most of which focus on recognizing machining features [1-3, 4, 9]. Although the term “feature” is domain-dependent and different names are used to represent features of particular domains, they can be classified into two basic categories - protrusions and depressions, regardless of domain of application [4, 5]. Therefore, feature recognition can be viewed as a technique to identify protrusions and depressions associated with unique geometric or functional characteristics related to a particular domain of interest.

There are mainly two approaches for feature recognition. One is the graph-based approach [1, 2, 6-8], of which face-adjacency-graph method [1, 7] is an example. In this approach the faces and edges of a solid model and their topological relationships are represented as graphs and they are searched to recognize subgraphs of unique patterns for features. A typical example is the inner loop of concave edges on a single face shown in Fig. 1(a), which is a hint of the existence of a protrusion feature. Similarly, an inner loop of convex edges on a single face is a hint of a depression feature. The graph-based approach has strength in recognizing isolated features with respect to performance and robustness, but it has difficulty in recognizing intersecting features since the unique patterns of features are destroyed when features intersect.

The other, the volumetric approach [3, 9-12], attempts to recognize features by generating volumes of features and mapping them onto the features of interest. ASVP (alternating sum of volumes with partitioning) [3] and MVD (maximal volume decomposition) [11] methods are examples. This approach has strength in recognizing intersecting features, but it has geometric restrictions, and combinatorial complexity may arise for complicated solid models.

Each method for feature recognition has pros and cons. It would be nice to develop a hybrid system of several methods by complementing the drawbacks of others. For example, a graph-based method is used to recognize and remove isolated features from a complex solid model and then a volumetric method can be applied to the simplified model to recognize intersecting features [13]. However, the types of isolated features that can be recognized from solid models by a graph-based method are usually limited to features on inner loops of single faces. For example, the simple protrusion feature in Fig. 1(b) is topologically similar to the one in Fig. 1(a), but it is not sure which are the protrusion faces because the loop is not an inner loop on a single face. If this kind of protrusion can be recognized without enumerating all the patterns of possible features, it would be helpful for enhancing the applicability of graph-based approach and could improve the efficiency of hybrid feature recognition.

*Corresponding author. Tel.: +82 2 760 4149, Fax.: +82 2 760 4329
E-mail address: yhwoo@hansung.ac.kr
† Recommended by Associate Editor Ki-Hoon Shin



(a) (b)

Fig. 1. Examples of protrusions: (a) protrusion on a single face; (b) protrusion on multiple faces.

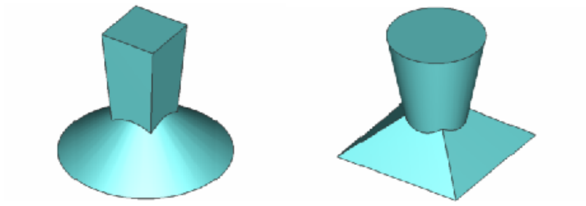
Inspired by this, we propose a quantitative measure that can be used in recognizing a protrusion of generic type bounded by a loop of concave edges in this paper. This measure is named as orthogonal bounding factor (OBF). The concept and formal definition of the orthogonal bounding factor is presented in the following section.

## 2. Orthogonal bounding factor

### *2.1 Background*

When people see an object, they identify a protrusion on the object by visually recognizing the area arising above the level of the surrounding surface. It becomes more distinctive as the protrusion stands more upright. In feature recognition by computer, however, protrusions are recognized by performing topological and geometric reasoning with the solid models. A loop of concave edges, whether it is on a single face or not, is a strong hint of a protrusion feature. If the loop of concave edges is an inner loop on a single face as shown in Fig. 1(a), it signifies the existence of a protrusion and the face(s) adjacent to the single face sharing the inner loop are the protrusion face(s). In such a case, the volume of the protrusion can be easily generated and removed from the model by simple geometric operations [14].

When a protrusion lies on multiple faces as shown in Fig. 1(b), the loop of concave edges bounding the protrusion cannot be judged whether it is an inner loop or not since it does not lie on a single face. This means that we still do not know which set of faces sharing the concave edges are the protrusion faces. Consequently, an additional criterion is needed for judging which faces are the protrusion faces. Yet topological reasoning does not seem promising for the judgment in general. For example, for the solid model in Fig. 2(a), the conical face is the one to be recognized as the protrusion face, while the four planar faces are the ones to be recognized as the protrusion faces for the solid model in Fig. 2(b). However, the two solid models are topologically identical and there is no general way of distinguishing them. Nonetheless, in each case the faces recognized as a protrusion stand more upright than the surrounding faces. That is, if a set of faces along a loop of concave edges are more upright than the other, it would be natural to recognize them as protrusion faces. This is the motivation for developing the orthogonal bounding factor, which can be used for a new measure of possibility of being a protrusion. In the following section, the formal definition of OBF is presented.



(a) (b)

Fig. 2. Which are the protrusion faces? The two solid models are topologically identical: (a) the conical face is the protrusion face; (b) the four planar faces are the protrusion faces.

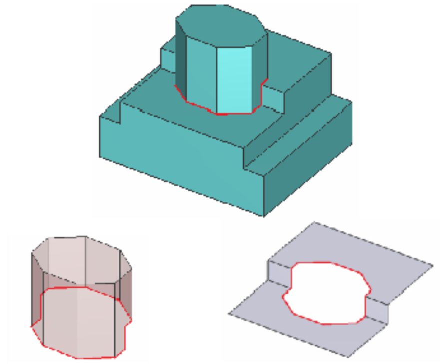


Fig. 3. The two sets of faces along a loop of concave edges.

### *2.2 Definition of orthogonal bounding factor*

In a manifold solid model, each edge is owned by exactly two faces. Therefore, the faces along a loop of concave edges can be grouped into two sets as shown in Fig. 3. In each set the face associated with the $i$-th edge of the loop comprising $n$ concave edges is denoted by $F_i$. Since a face can own several edges of the loop, there may be a case where $F_i$ and $F_{i+1}$ are the same. Now the underlying curve for the $i$-th edge of the loop is denoted by $\mathbf{r}_i(t)$, where $t_i^s \le t \le t_i^e$. Here $t_i^s$ and $t_i^e$ are the starting and ending parameters for the $i$-th edge, respectively. Accordingly, $\mathbf{r}_i(t_i^s)$ and $\mathbf{r}_i(t_i^e)$ are the starting and end points of $i$-th edge, respectively. Note that $\mathbf{r}_i(t_i^e) = \mathbf{r}_{i+1}(t_{i+1}^s)$ and $\mathbf{r}_n(t_n^e) = \mathbf{r}_1(t_1^s)$.

Let $\mathbf{N}_i(\mathbf{r}_i(t))$ denote a vector normal to $F_i$ at a point on the $i$-th edge. Now, when the loop has $n$ concave edges and each edge of the loop is discretized into $m+1$ equally-distributed points, a vector normal to $F_i$ at the $j$-th point of the $i$-th edge is defined as follows:

$$\mathbf{N}_{i,j} = \mathbf{N}_i\big(\mathbf{r}_i(t_i^s + j\Delta t_i)\big),\ i = 1,\cdots,n, j = 0,\cdots,m.$$
$$\Delta t_i = (t_i^e - t_i^s)/m. \tag{1}$$

Now, $\mathbf{W}(m)$ is defined in Eq. (2) as a vector function of variable $m$, which is the sum of the cross product of two

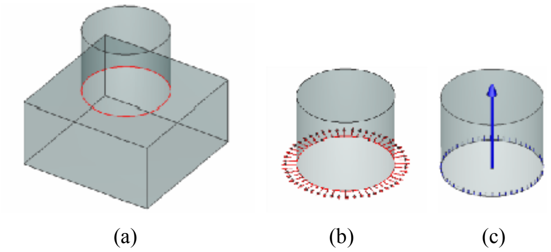



Fig. 4. An example of OBF calculation: (a) a cylindrical protrusion bounded by a loop of concave circular edge of radius R; (b) The red arrows depict the normal vectors at the points on the cylindrical face along the loop of concave circular edge; (c) The small blue lines depict $\mathbf{N}_{1,j} \times \mathbf{N}_{1,j+1}$ and the big blue arrow the $\mathbf{W}(m)$ .

consecutive normal vectors.

$$\mathbf{W}(m) = \textstyle\sum_{i=1}^{n} \sum_{j=0}^{m} \left(\mathbf{N}_{i,j} \times \mathbf{N}_{i,j+1}\right), \quad (2)$$

where $\mathbf{N}_{i,m+1} = \mathbf{N}_{i+1,0}$ *and* $\mathbf{N}_{n,m+1} = \mathbf{N}_{1,0}$.

Finally, OBF is defined in Eq. (3) as the norm of $\mathbf{W}(m)$ with the infinite value of $m$ .

$$\text{OBF} = |\lim_{m\to\infty} \mathbf{W}(m)|. \quad (3)$$

The value of OBF increases as more $\mathbf{N}_{i,j}$ are orthogonal to a common direction. Theoretically, the maximum value of OBF for a convex protrusion is $2\pi$ .

OBF for the cylindrical face along a circular concave edge shown in Fig. 4 is derived, for example. The concave circular edge is represented as $\mathbf{r}(t) = R\cos(t)\boldsymbol{u} + R\sin(t)\boldsymbol{v}$, where $0 \le t \le 2\pi$. Here $\boldsymbol{u}$ and $\boldsymbol{v}$ denote the major and the minor axes of the circular edge, respectively. The vector $\mathbf{N}(\mathbf{r}(t))$ normal to the cylindrical face at a point on the circular edge is then:

$$\mathbf{N}(\mathbf{r}(t)) = \cos(t)\boldsymbol{u} + \sin(t)\boldsymbol{v},$$

$$\mathbf{N}_{i,j} = \mathbf{N}(\mathbf{r}(j\Delta t)),\ i = 1,\ j = 0,\cdots,m. \quad \Delta t = 2\pi/m.$$
$$\mathbf{N}_{1,j} = \cos(j\Delta t)\boldsymbol{u} + \sin(j\Delta t)\boldsymbol{v},$$

$$\begin{aligned}\mathbf{N}_{1,j} \times \mathbf{N}_{1,j+1} &= \{\cos(j\Delta t)\boldsymbol{u} + \sin(j\Delta t)\boldsymbol{v}\} \\ &\quad \times \{\cos((j+1)\Delta t)\boldsymbol{u} + \sin((j+1)\Delta t)\boldsymbol{v}\}.\end{aligned}$$

Using the trigonometric formulas, $\cos((j+1)\Delta t) = \cos(j\Delta t)\cos\Delta t - \sin(j\Delta t)\sin\Delta t$ and $\sin((j+1)\Delta t) = \sin(j\Delta t)\cos\Delta t + \cos(j\Delta t)\sin\Delta t$, and letting $\boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}$, the above equation becomes:

$$\begin{aligned}&\mathbf{N}_{1,j} \times \mathbf{N}_{1,j+1} \\ &= \cos\Delta t\{\cos(j\Delta t)\sin(j\Delta t) - \cos(j\Delta t)\sin(j\Delta t)\}\boldsymbol{w} \\ &+ \sin\Delta t\{\cos(j\Delta t)\cos(j\Delta t) + \sin(j\Delta t)\sin(j\Delta t)\}\boldsymbol{w} \\ &= \sin\Delta t\boldsymbol{w},\end{aligned}$$

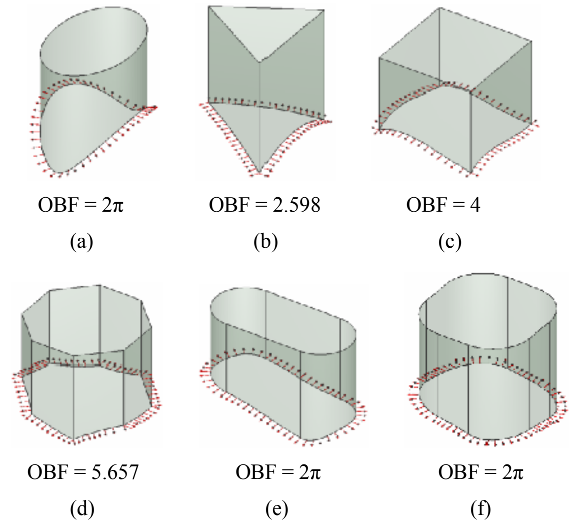



Fig. 5. OBFs of some typical protrusion features: (a) an elliptical protrusion; (b) a regular triangular protrusion; (c) a rectangular protrusion; (d) an octagonal protrusion; (e) a protrusion with two convex cylindrical faces; (f) a rectangular protrusion with rounded corners.

$$\mathbf{W}(m) = \sum_{i=1}^{1}\sum_{j=0}^{m} \sin\Delta t\boldsymbol{w} = (m+1)\sin\Delta t\boldsymbol{w},$$

$$\begin{aligned}\text{OBF} &= \left|\lim_{\Delta t\to 0}(m+1)\sin\Delta t\boldsymbol{w}\right| \\ &= \lim_{\Delta t\to 0}\left(\frac{2\pi}{\Delta t}\sin\Delta t\right) \\ &= 2\pi\lim_{\Delta t\to 0}\left(\frac{\sin\Delta t}{\Delta t}\right) = 2\pi.\end{aligned}$$

An interesting and useful property of OBF is that it is invariant for a protrusion. For example, OBF for a full cylindrical face along any loop of edges containing the axis of the face is always $2^{\pi}$ regardless of the shape of the loop. OBFs for some typical protrusions are shown in Fig. 5. The small arrows in the figure depict the vectors normal to the faces along the loop of edges. In Fig. 6, the OBF decreases as the rectangular protrusion tapers.

## 3. Protrusion recognition using OBF

First, in order to recognize protrusions from solid models using OBF, the loops of concave edges on multiple faces (LMF) should be identified. As compared to the identification of a loop of concave edges on a single face (LSF), the identification of LMF is not straightforward. Since an LMF spans over several faces, consideration of the topology of a single face does not help. Instead, the connecting relationship among all the concave edges needs to be considered. To achieve this, we adopted graph theory. In fact, commercial geometric modeling kernels provide classes and procedures for the graph theory. For example, the ACIS geometric modeling kernel of Spatial Corporation provides *generic_graph* class that includes the member functions for generating graphs from lists

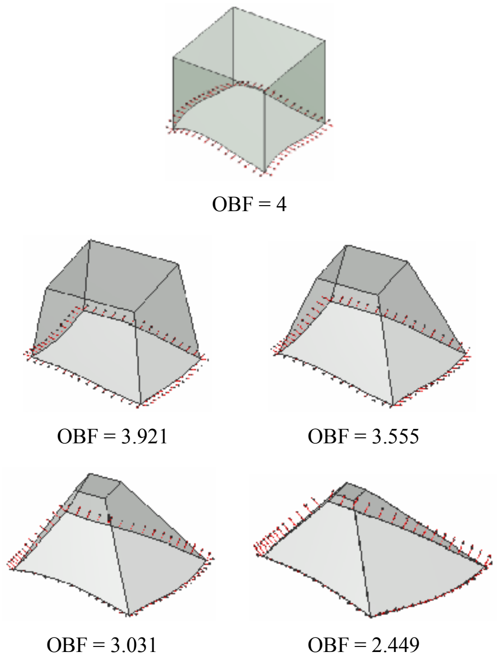
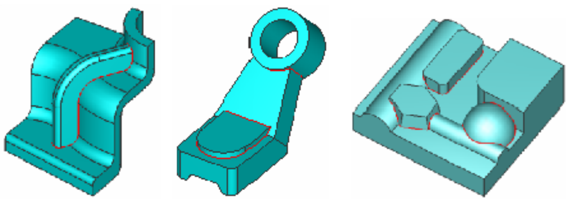



Fig. 6. The OBF decreases as the rectangular protrusion tapers.



(a) (b) (c)

Fig. 7. LMFs (in red) identified by *is_cycle*() method of the ACIS *generic_graph* class.

of faces and edges and finding cyclic loops from the graphs. By utilizing the member function *is_cycle*() of the ACIS *generic_graph* class, LMFs of a solid model are identified in this paper. Fig. 7 shows some examples of LMFs identified by the method.

The process of recognizing protrusions is shown in Fig. 8. Given a loop of concave edges identified by the method of graph theory, the faces owning the edges are collected and grouped into two sets S1 and S2. Then OBF for each set is calculated as shown in Fig. 9. The variable $m$ in Eq. (3) approaches infinity, but it is set to a reasonable value in actual implementation ($m$ = 50). If OBF for the face set S1 is bigger than that for the face set S2, The faces in S1 can be recognized as protrusion faces.

However, as shown in Fig. 9(b), it occurs that OBF for the faces of a depression is also bigger than that for the surrounding face. Therefore, it is necessary to distinguish depressions from protrusions and the directionality of faces is taken into account in order to address the problem. To determine the directionality of faces in a set, the midpoint of the bounding

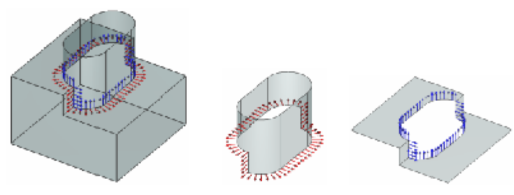
```
int Recognize_protrusion (
input: A loop of concave edges (LCE),
output: A list of protrusion faces (PFL)
)
{
    Initialize PFL;
    Get the faces owning the edges of LCE;
    Classify the faces into two sets of faces S1 and S2
      based on the adjacency;

    Calculate OBF1, the OBF for S1;
    Calculate OBF2, the OBF for S2;

    Determine D1, the directionality of the faces in S1;
    Determine D2, the directionality of the faces in S2;

    if(OBF1 > OBF2   &&   D1 == OUTWARD) {
      PFL = S1;
      return 1;
    }
    else if(OBF2 > OBF1   &&   D2 == OUTWARD) {
      PFL = S2;
      return 1;
    }
    else {
       return 0;
    }
```

Fig. 8. The process of recognizing a protrusion using OBF.



(a)

(b)

Fig. 9. Calculation of OBF for the set of faces along the loop of concave edges: the blue and red arrows depict the normal vectors: (a) protrusion; (b) depression.

box of the loop of concave edges is found. If the vectors normal to the faces point to the midpoint, the directionality is inward, otherwise it is outward. This can be achieved by taking the dot product of the normal vector and the position vector of the mid-point. If a set of faces has a bigger value of OBF than the other and the directionality is outward, the faces

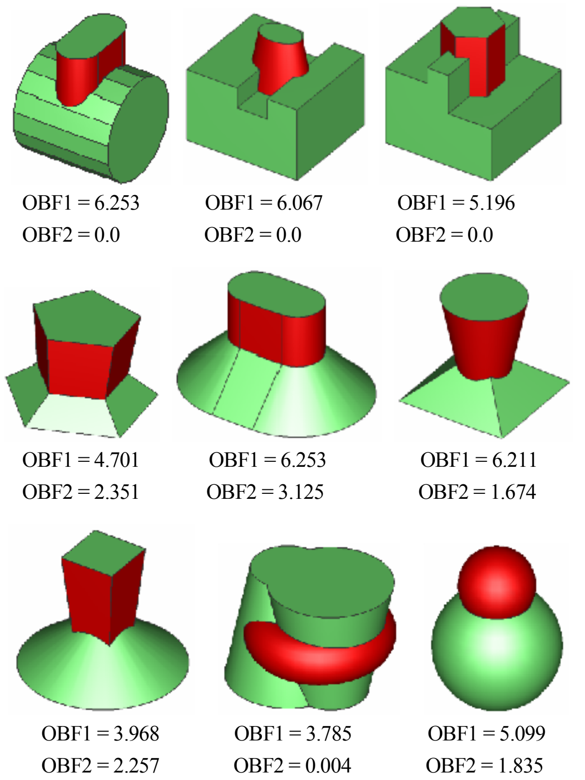



Fig. 10. Examples of the protrusions recognized by using OBF: faces in red are the protrusion faces recognized by the method shown in Fig. 8. OBF1 denotes OBF for the protrusion faces and OBF2 the one for the other set of faces sharing the loop of concave edges.

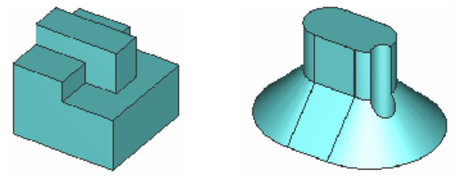


Fig. 11. Examples of protrusions that cannot be recognized by the method.

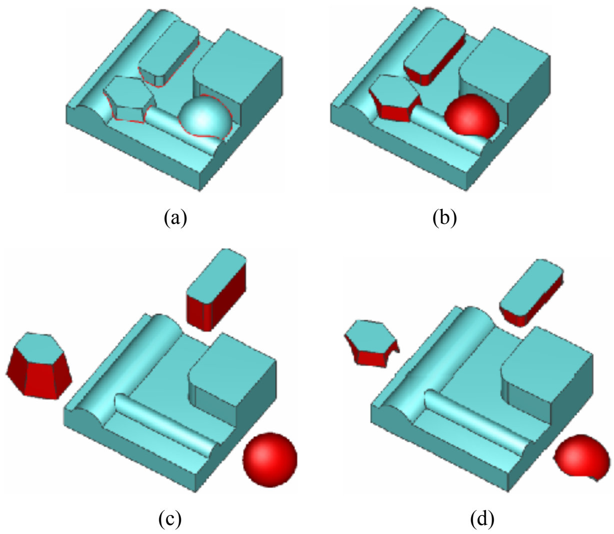



Fig. 12. Protrusion recognition for an example model: (a) the identified loops of concave edges; (b) the protrusion faces recognized using OBF; (c) selective volume decomposition with the protrusion faces; (d) the net volumes of the protrusion features.

in the set are recognized as protrusion faces.

The method presented in this paper has been implemented as a system using C/C++ with the ACIS geometric modeling kernel running on a PC. Fig. 10 shows the examples of protrusion faces recognized by the method. Delta volume of a solid model is defined as the difference between the solid model and the bounding box of the solid model. A depression feature in a solid model appears as a protrusion in the delta volume of the solid model. Therefore, it is claimed that the method can be applied to recognizing depression features.

## 4. Limitations and future work

A novel method for recognizing protrusions of generic type using the orthogonal bounding factor has been presented in this paper. Examples have been also shown to demonstrate the effectiveness of the method for feature recognition. The contributions of the research are summarized as follows:

- The concept of the orthogonal bounding factor has been introduced and its formal definition has been established.
- OBF can be used as a quantitative measure of the possibility of being a protrusion for a given loop of concave edges.
- There is no need to enumerate the types of all the possible features since the method does not require predefined patterns for protrusion types.

However, there are also limitations of this method and the some of them are summarized as follows:

- Calculation of OBF is limited to the protrusions that have a loop of concave edges. If a protrusion is not bounded by a loop of concave edges as shown in Fig. 11, it cannot be recognized.
- Identification of loops of concave edges from a solid model is crucial for the method. Though some commercial geometric modeling kernels provide methods for identifying loop of concave edges, it is not robust enough for a complex model where some loops are coupled and overlap. A more robust and complete method for identification of such loops is needed.
- OBF is a supplemental measure for graph-based feature recognition. It cannot be used as the sole criterion for determining a protrusion feature.

Once the faces of a protrusion feature are recognized, the volume of the feature needs to be generated from the faces so that it can be removed from the solid model. Therefore, for the method presented in this paper to be more practical and effective, a technique to generate the volume of feature from recognized protrusion faces is needed. A possible suggestion is the use of selective volume decomposition [15] as shown in Fig. 12. Generation of volumes from the recognized protrusion faces will be a major future work associated with this research.

Finally, as a new method, it may not be perfectly general to accommodate every kind of real industrial model and there may exist some limitations that as of yet have not been discovered. However, this research is not final and we will continue to enhance the method as the research goes on.

## Acknowledgment

This research was financially supported by Hansung University.

## References

[1] S Joshi and TC Chang, Graph-based heuristics for recognition of machined features from a 3D solid model, *Computer-Aided Design*, 20 (2) (1988) 58-66.

[2] J Corney and DER Clark, Method for finding holes and pockets that connect multiple faces in 2 1/2D objects, *Computer-Aided Design*, 30 (7) (1998) 658-668.

[3] DL Waco and YS Kim, Geometric reasoning for machining features using convex decomposition, *Computer-Aided Design*, 26 (6) (1994) 477-489.

[4] H Sakurai and C Chin, Defining and recognizing cavity and protrusion by volumes, *Proceedings of Computers in Engineering*, ASME (1993).

[5] SH Chuang, YC Tsai, CY Du and CS Yang, Decomposing solid models into depression and protrusion features, *International Journal of Integrated Manufacturing*, 8 (6) (1995) 393-398.

[6] LK Kyprianou, Shape classification in computer aided design, *Ph.D. Thesis*, Christ College, University of Cambridge, UK (1983).

[7] E Bruzzone and LD Floriani, Extracting adjacency relationships from a modular boundary model, *Computer-Aided Design*, 23 (5) (1991) 344-355.

[8] T Lim, H Medillin, C Torres-Sanchez, JR Corney, JM Ritchie and JBC Davies, Edge-based identification of DP-features on free-form solids, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 (6) (2005) 851-860.

[9] Y Woo and H Sakurai, Recognition of maximal features by recursive volume decomposition, *Computer-Aided Design*, 35 (11) (2003) 969-977.

[10] Y Woo, Fast cell-based decomposition and applications to solid modeling, *Computer-Aided Design*, 35 (11) (2003) 969-977.

[11] H Sakurai and P Dave, Volume decomposition and feature recognition, Part II: curved objects, *Computer-Aided Design*, 28 (6-7) (1993) 519-537.

[12] D Sandiford and S Hinduja, Construction of feature volume using intersection of adjacent surfaces, *Computer-Aided Design*, 33 (2001) 455-473.

[13] Y Woo, E Wang, YS Kim and HM Rho, A hybrid feature recognizer for machining process planning system, *Annals of the CIRP*, 54 (1) (2005) 397-400.

[14] S Koo and K Lee, Wrap-around operation to make multi-resolution model of part and assembly, *Computers and Graphics*, 26 (2002) 687-700.

[15] Y Woo and SH Lee, Volumetric modification of solid CAD models independent of design features, *Advances in Engineering Software*, 37 (2006) 826-835.

**Yoonhwan Woo** is currently an associate professor of Mechanical Systems Engineering in Hansung University in Korea. His Ph.D. in Mechanical Engineering is from Colorado State University, the M.S. degree in Mechanical and Aerospace engineering from Illinois Institute of Technology, and the B.S. degree in Precision Mechanical Engineering from Hanyang University in Korea. His research interests include solid modeling, CAD model idealization for analysis, and computer aided process planning.

**Sang-Hyun Kim** obtained his B.S. (1993) and M.S. degrees (1995) in Aerospace Engineering from Pusan Nation University and Ph.D. (2005) in Aerospace Engineering from Texas A&M University. He is currently a faculty member in Mechanical Systems Engineering at Hansung University. His research interest lies in design, fabrication and test of MEMS/NEMS, especially encapsulation of liquid in porous nanofibers. He is also interested in the development of design platform of multi-physics and multiscale phenomena.