Proceedings of DETC'01
ASME2001 Design Engineering Technical Conferences
Semptember 9-12, 2001

DETC2001/DAC-21010

SHEET MODELING AND THICKENING OPERATIONS BASED ON NON-MANIFOLD BOUNDARY REPRESENTATIONS

Sang Hun Lee
Graduate School of Automotive Engineering
Kookmin University
Jeongneung-Dong,Seongbuk-Gu,Seoul
136-702,Korea
E-mail : shlee@kookmin.ac.kr

Hyun-Soo Kim
SAMSUNG SDS Co. LTD
530, Jangpyung-Ri, Sinhyun-Up, Koje-Si
Kyungnam, 656-710, Korea
E-mail : kdzmesnb@hanmail.net

ABSTRACT

This paper describes sheet modeling and thickening operations based on a non-manifold topological representation for efficient solid modeling of thin-walled plastic or sheet metal parts. Since the existing methods have adopted boundary representations for solid models, it is difficult to represent the exact adjacency relations between topological entities in a sheet model, and to describe a mixture of wireframe and sheet objects that appear in the intermediate steps of sheet modeling operations. Accordingly, it is difficult to devise and implement the algorithms for sheet modeling and thickening operations. To solve these problems, we introduce a non-manifold boundary representation as a topological framework and propose a sheet thickening algorithm by presenting variations to a general non-manifold offset algorithm that is based on the mathematical definition of offsets. In addition, to facilitate sheet modeling operations, not only a set of generalized Euler operators for non-manifold models are provided, but also sheet modeling capabilities, including adding, bending, and punching functions with two-dimensional curve editors.

Key Words : *CAD, Sheet Model, Non-manifold, Offset, Thickening*

1. INTRODUCTION

Plastic and sheet metal parts, both thin and constant thickness, have been gaining wider usage in industry, as they contribute to reducing the weight of a product. Recently, as solid modeling systems are used more widely in mechanical design, the efficiency of thin-walled part modeling capabilities has become more important [1, 2, 3, 4]. There have been several research studies based on increasing efficiency and they can be classified into two categories: solid shelling methods and sheet thickening methods. As illustrated in Figure 1, in solid shelling methods [5, 6], a thin-walled solid model is generated by digging out the inside volume from the solid model of the outside shape. However, in sheet thickening methods [1, 2, 3], as shown in Figure 1(b), a sheet model is created for one side or a medial surface of the part first, and then a thin-walled solid is generated by adding volume to a sheet by a given thickness. To facilitate the discussion, let us define a sharp edge as an edge that is adjacent to only one face and constitutes the boundary of a sheet, and define a thickness face as a face that is a thin strip-like face connecting the inside and the outside walls. In sheet thickening approaches, the sharp edges in a sheet body are converted to the thickness faces.

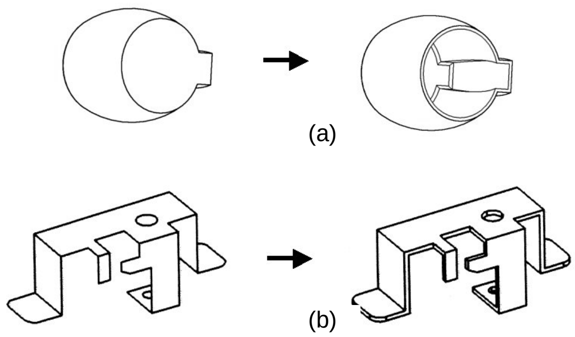


Figure 1. Thin-walled solid modeling methods: (a) solid shelling; (b) sheet thickening.

Lee and Lee [5] first developed solid shelling capabilities using a commercial solid modeling kernel, ROMULUS. In their algorithm, a solid object for an outer or inner wall is modeled first. Then, another solid for the other side of the wall is created by copying the original solid and then shrinking the copied solid by a given thickness. Finally, a thin-walled solid is generated by subtracting the offset solid from the original solid using the Boolean operations. In this method, however, the authors did not consider any correction process for illegal topology caused by self-intersection in the shrinking step.

Forsyth [6] also proposed an algorithm for shelling operations on B-rep solids, and implemented the algorithm with the modeling capabilities of SolidDesigner. His algorithm is the same as that of Lee and Lee [5] except that it checks to correct self-intersection of the shelled body and gives rounding to convex edges and vertices of the offset solid.

Although these shelling operations are very useful for modeling plastic parts in the shape of bowls, they are not suitable for modeling sheet metal parts with many bending and hole features. To solve this problem, research on sheet thickening methods has been carried out by several researchers. In most of the existing algorithms, a sheet object is modeled as a degenerate solid with zero thickness, and then it is converted to a solid by replacing its geometry with the offset one.

Stroud [1] suggested a method to convert a sheet into a solid for a given thickness. In his system, a sheet object is represented as a degenerate B-rep solid model, in which the thickness faces are represented by sharp edges. In the transformation procedure, the sharp edges and their end vertices are first split in order to make topological data for the thickness faces. Then the geometry for each vertex, edge, and face is calculated and assigned to the corresponding topological entity. However, this method may result in unacceptable, impractical solids, as he did not suggest any checking and correction methods for self-intersections of the converted solid.

Lee and Kwon [2] proposed another sheet modeling and transformation approach, which was revised by Lim and Lee [3]. They adopt the winged edge data structure as a topological framework for representing sheet objects. From a schematic viewpoint, this approach is very similar to Stroud's, as they both adopt solid data structures in order to represent sheet objects, which are regarded as degenerated solids. However, there is a difference between the two approaches in storing topological data for the degenerated solids. In Lee and Kwon's work, a sheet model has full topological data for the corresponding solid model, including the thickness faces, whereas in Stroud's work, the thickness faces are degenerated into sharp edges. Therefore, in Lee and Kwon's method, sheet thickening is basically just replacing the geometry of each topological entity with the offset geometry. Of course, however, this method may cause unacceptable, impractical solid models, as the topology of the acceptable solid is not coincident with the topology of the sheet body, and self-intersection of the offset solid is not considered. To complement these drawbacks, they investigated failure cases and suggested topological correction methods for unacceptable solids, although the cases are limited; nor is the self-intersection problem considered. On the other hand, in order to facilitate the development of high-level sheet modeling capabilities, they also proposed a set of sheet Euler operators that are like macros of the standard Euler operators used for solid modeling. However, as their sheet Euler operators are not a complete set of topological operators for manipulating topological entities of a sheet model, the standard Euler operators must still be used together.

The methods of Stroud [1] and Lee and Kwon [2] have a common problem in that their topological data structures do not store and provide proper information about the adjacency relationships of the topological entities. This is because sheet objects are described by a solid data structure, while sheets are fundamentally non-manifold objects. This deficiency makes it difficult to develop algorithms for sheet modeling and thickening capabilities. Although the existing commercial non-manifold modeling systems usually provide shelling and sheet thickening capabilities, we are unaware of any reported research work on their algorithms to date.

Recently, Lee [7] proposed offsetting operations on three-dimensional non-manifold B-rep models that include wireframes, sheets, solids, and their mixtures. Lee's algorithm corrects global self-intersections of the thickened body automatically, whereas local self-intersections of each face are not yet corrected. Using this method, a sheet can be thickened in a universal way without considering any exceptional cases in the generation of thickness faces. Unfortunately, however, the thickness faces of the resulting body always have tubular surfaces, as the algorithm is based on the mathematical definition of offsets, which is that the boundary of the offset body lies at an offset distance from the original sheet. Such tubular thickness faces are rarely used in actual plastic and sheet metal parts. Therefore, it is necessary to modify this algorithm if one wanted to use it for modeling of thin-walled parts.

In this paper, we suggest a sheet modeling and thickening method on the basis of a non-manifold boundary representation as a topological framework. Briefly, we adapt the non-manifold offset algorithms proposed by Lee [7] for practical sheet thickening operations. By adopting a non-manifold representation, all problems of the previous sheet modeling and thickening methods based on solid data structures are expected to be solved, as a non-manifold data structure can represent solids, sheets, wireframes, and their mixtures in a single framework. That is, all of the correct topological information can be provided by interrogation functions, any macros called by the sheet Euler operators do not need to be developed in virtue of a complete set of non-manifold Euler operators, and the sheet thickening algorithm can be devised in an easier and more comprehensive way.

The rest of this paper is composed as follows: Section 2 briefly describes the system architecture and data structure adopted in this work. Section 3 introduces a set of Euler operators and sheet modeling functions that manipulate sheet models in a non-manifold modeling environment. Sections 4, 5, and 6 describe our algorithms for converting a sheet into a solid; these are variations of the general non-manifold offset algorithms proposed by Lee [7]. Section 7 shows a case study. Section 8 discusses some exceptional cases of automatic sheet thickening process. In Section 9, we present our conclusions.

2. NON-MANIFOLD GEOMETRIC MODELING SYSTEM

To date there have been many representation schemes proposed for non-manifold topology such as the radial edge structure [8], the vertex-based B-rep [9], the selective geometry complex [10], the partial entity structure [7, 11], and so on. In this paper, we adopt the partial entity structure proposed by Lee and Lee [11] as a topological data structure for non-manifold models.

In this work, ANYSHAPE is used as a kernel engine for non-manifold geometric modeling [11]. It has been developed by the authors, based on partial entity structure and object-oriented programming technology. All of the information on the geometric or topological structures of the kernel, and all operations on them are provided as a form of class libraries in C++ as illustrated in Figure 2.

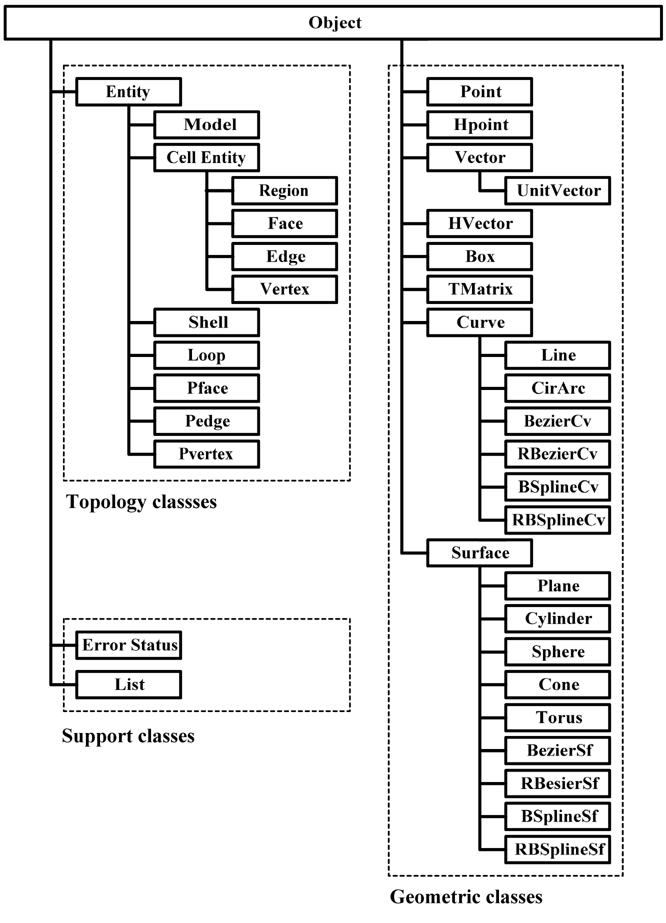


Figure 2. Class hierarchy of our non-manifold modeling kernel

The kernel is composed of the geometric engine, the topological engine, the application procedure interface (API) and the supporting module. The geometric engine includes the classes for points, curves and surfaces, and their member functions for vector and matrix operations, interrogation and manipulation of curves and surfaces, and intersection between curve/surface and curve/surface. The topological engine comprises the classes for topological entities and their member functions for topological interrogation, Euler operators, and high-level modeling operations such as primitive generation, local modification, sweeping and the Boolean operations. The supporting module provides supplementary modeling facilities such as journalizing and error recovery. In addition, the APIs are a collection of interface routines to the kernel functionality, by which the users can access and utilize the modeling operations and facilities conveniently. Therefore, all applications of this kernel system can access the kernel directly, through the class libraries as well as through APIs.

3. SHEET MODELING

Just as in Stroud's [1] and Lee and Kwon's [2] works, our system provides users not only with high-level modeling functions for creating and modifying sheet models, but also with such low-level functions as non-manifold Euler operators. To facilitate usage of these functions, interactive graphical user interfaces and 2-D curve editors are provided in our system. The rest of this section briefly explains the non-manifold Euler operators and the high-level sheet modeling functions.

3.1. Non-Manifold Euler Operators

Most B-rep solid modeling systems use Euler operators, as they guarantee the topological integrity of the models and facilitate the implementation of high-level modeling functions. Several researchers such as Masuda [12] and Yamaguchi and Kimura [13] have derived proper Euler formulas applicable to 3-D non-manifold models and proposed a set of Euler operators based on the derived Euler formula. In this paper, we accepted the Euler formula proposed by Yamaguchi and Kimura [13] as follows:

$$V - E + F - L = S - C + R$$

where V, E, F, and L on the left side are the number of vertices, edges, faces, and hole loops, respectively, and S, C, and R on the right side are the number of void shells, non-manifold cycles, and regions, respectively. From the above equation, we can see that the six independent Euler operators and their six inverse operators are sufficient to manipulate the topological data of non-manifold models. To facilitate the development of high-level modeling functions, however, we added eight Euler operators, and two non-Euler topological operators for initially creating and finally deleting a model. The basic and extended Euler operators used in our system are listed in Lee and Lee's paper [11].

An example of a sheet modeling process using the Euler operators is shown in Figure 3. In this way, users can create and modify any shape of sheet objects interactively using our Euler operators directly, whereas the macro sheet Euler operators proposed by Lee and Kwon [2] are not sufficient to generate an arbitrary sheet model.

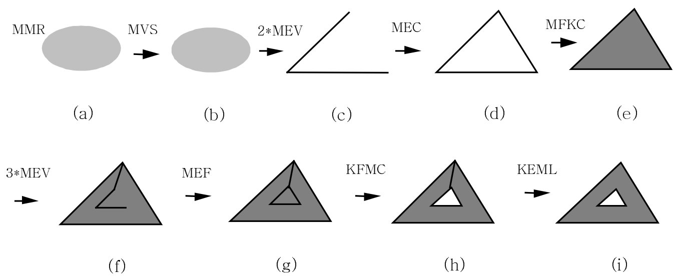
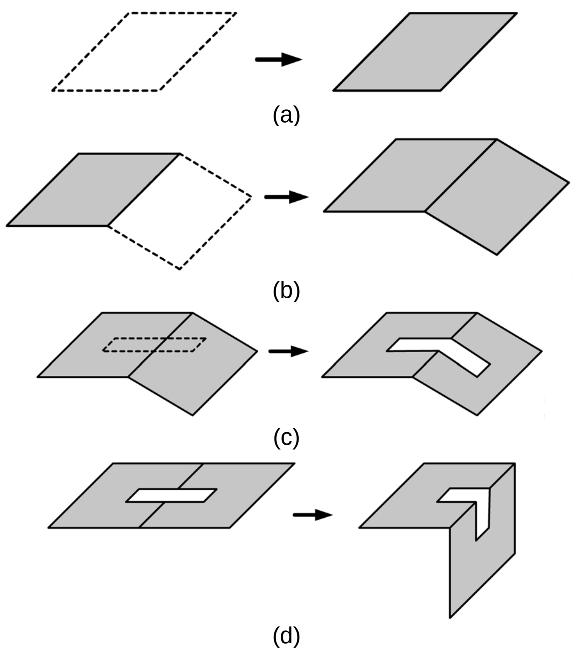


Figure 3. Example of sheet modeling with the generalized Euler operators

3.2. Sheet Modeling Capabilities

Although non-manifold Euler operators allow the creation of any shape of sheet model, their modeling productivity is not high, as they are basically low-level modeling functions. Thus, it is necessary to provide high-level modeling functions, which are implemented with a series of Euler operators. As shown in Figure 6, this system offers the following modeling functions in menus, combined with two-dimensional curve editors for interactive use. The dotted lines in Figure 4 are input curves created by the interactive curve editors.

- CREATE: creating a sheet from a drawing
- ADD: adding a face to a sheet
- PUNCH: punching a sheet to create a hole
- BEND: bending a sheet with respect to an edge
- SWEEP: sweeping an edge to create a face
- DRAW: drawing a part of sheet
- ROUND: rounding sharp edges



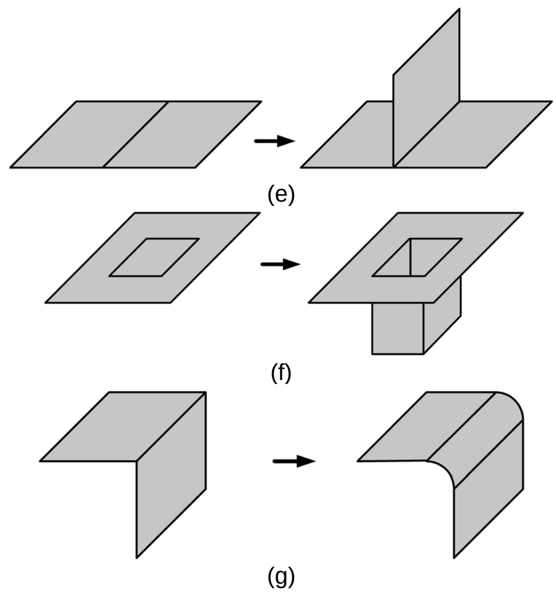


Figure 4. Sheet modeling capabilities: (a) CREATE; (b) ADD; (c) PUNCH; (d) BEND; (e) SWEEP; (f) DRAW;(g) ROUND

4. OVERVIEW OF SHEET THICKENING ALGORITHM

According to the mathematical definition for offsetting a non-manifold model *M*, the positive offset of a sheet model by a distance *d* can be understood as the volume swept by a solid sphere of radius *d* as its center moves throughout *M*. This definition is equivalent to the Minkowski sum of a solid sphere and the original model. If the positive offset algorithm proposed by Lee [7] is applied to an L-shaped sheet model shown in Figure 5(a), the offset solid of a sheet has thickness faces with tubular surfaces to satisfy the mathematical definition of offsets as shown in Figure 5(b). However, tubular thickness faces are rarely shown in actual plastic or sheet metal parts. Therefore, we need to modify Lee's algorithm to have the practical thickness faces generate an offset solid as shown in Figure 5(c).

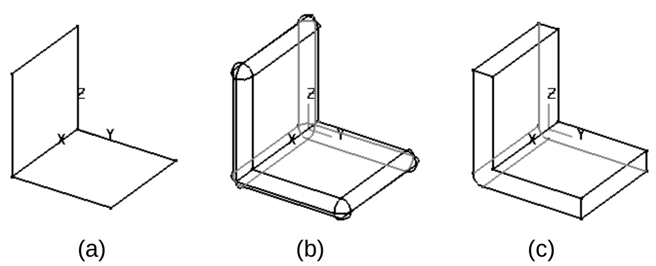


Figure 5. Comparison of the mathematical offset model with the practical one: (a) a simple L-shaped sheet model; (b) a solid model constructed by an offset algorithm based on a mathematical definition of offsets; (c) an offset solid model as desired for practical part design

To satisfy this requirement, we propose a practical sheet offset algorithm for efficient solid modeling of thin-walled parts. As shown in Figure 6, this algorithm is composed of the following five steps:

(Step 1) Select all faces, edges, and vertices whose offsets will contribute to the construction of a superset of the boundary of the offset model of *M*.
(Step 2)Generate offset models for the faces, edges, and vertices selected in Step 1. In this step, the offset models for sharp edges are generated with ruled surfaces instead of tubular surfaces. In addition, the offset models for the vertices on the sharp edges are not generated, whereas in the mathematical algorithm they are generated with spherical surfaces.
(Step 3) Unite all of the offset models generated in Step 2 in order to create a superset model M_s whose boundary is $\partial(\partial M \oplus d)$.
(Step 4) Fill the holes of the thickness faces with appropriate surfaces. As the offset models of the vertices on the sharp edges have not been generated in Step 2, some holes may occur on the thickness faces after union operations. In this step, the system detects and closes these holes with a planar or freeform surface. This step is essential for the practical offset algorithm, whereas it is not necessary for the mathematical offset algorithm, as no hole occurs anywhere by generating the offset models for all of the vertices.
(Step 5) Remove unnecessary topological entities such as laminar faces, dangling edges, and isolated vertices. In the mathematical algorithm, all topological entities of M_s that are within the offset distance d from the boundary of the original model M, are detected and removed to obtain the exact offset model M_o. However, this method cannot be applied to the practical offset model, as the thickness faces are within d from ∂M. This step is realized with a different algorithm from the mathematical one.

As a result, comparing with the mathematical offset algorithm, Step 1 is identical, Step 4 is added, and Steps 2, 3, and 5 are modified to generate practical thin-walled solids.

The offset direction can be outside, inside, or in both directions of a sheet. If a sheet is modeled for outside wall shape of the part, it is offset inside. If it is modeled for inside wall shape, it is offset outside. If it is modeled for medial surface shape, it is offset in both inside and outside directions. These three cases share the same offset procedure above, except for small changes in Steps 1 and 2. Therefore, here we firstly explain the algorithm for offsetting in both inside and outside directions using the example model in Figure 7, then describe how to adapt this algorithm for offsetting operations in an inside or outside direction.

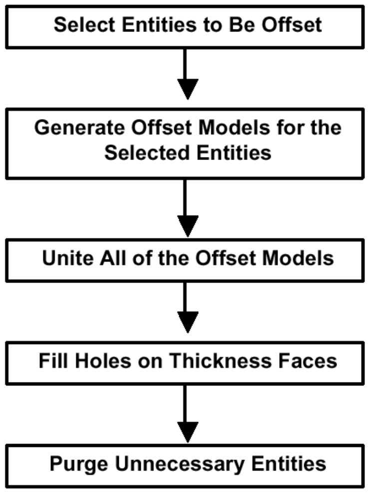


Figure 6. Sheet offset algorithm for creating practical thin-wall solid models

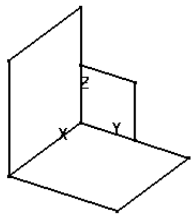


Figure 7. An example sheet model

5. SHEET THICKENING IN BOTH INSIDE AND OUTSIDE DIRECTIONS

5.1. Selection of Topological Entities to Be Offset

According to Lee [7], all of the faces, edges and vertices adjacent to the shells of the void regions of a given sheet model *M* are selected in this step, as the offsets of these entities participate in the boundary of the offset model of *M*. However, in the practical offset algorithm, the auxiliary topological entities such as face-uses, wedges, and disks, are generated and selected instead of faces, edges, and vertices, in order to facilitate the implementation of the latter steps. The auxiliary entities are classified into a few groups according to their convexity. In addition, the sharp edges are searched for special treatments in the next step. The definitions of the auxiliary entities are as follows.

- A face-use is a side of a face and constitutes a shell. As shown in Figure 8(a), its normal is directed to the inside of the adjacent region. In the partial entity structure, the partial face corresponds to the face-use.
- A wedge represents a corner space around an edge bounded by two adjacent face-uses. As shown in Figure 8(b), the edge E_1 has three wedges W_1, W_2, and W_3. Actually, the wedge class in our program stores three

pointers to the edge and two partial faces adjacent to the edge. For example, the wedge W_1 stores the pointers to three entities (E_1, PF_1, PF_2). Wedges are classified into convex, concave, and smooth. Here, smooth means satisfying the G1 continuity.

- A disk represents a corner space at a vertex bounded by its adjacent face-uses. Actually, the disk class in our program contains the partial vertex pointer and a list of wedge pointers ordered anti-clockwise. Disks are classified into convex, concave, smooth, and mixed, to facilitate the generation of the offset models of disks. For example, the disk D_1 in Figure 8(c) stores (*PV1*, W_4, W_5, W_6) and is classified as a convex disk.

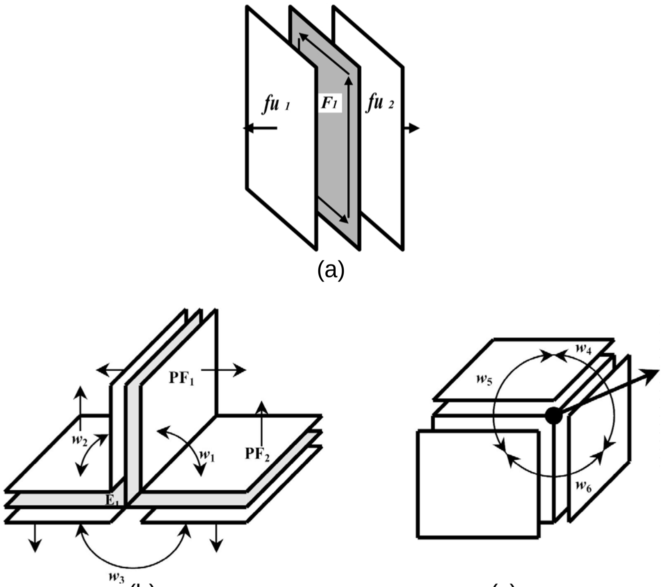


Figure 8. Auxiliary topological entities for sheet offsetting:
(a) face-uses; (b) wedges; (c) disks

5.2. Generation of Offset Models for the Selected Entities

In this step, the system generates offset models for all of the face-uses, wedges, and disks selected in the previous step. According to the mathematical definition of non-manifold offsets, the offset of a face-use is a normal offset patch which is obtained by offsetting a face by a given distance *d* along the face normal. The offset of a wedge is a tubular patch of radius *d* obtained by sweeping a disk along the edge curve as a trajectory. The offset of a disk by a distance *d* is defined as a spherical patch of radius *d* centered at the vertex position. Therefore, if this offset method is applied to a sheet, the thickness faces have tubular surfaces. Thus, the modification of the mathematical offset algorithm is essential for the generation of acceptable thin-walled solids.

In our approach, if a wedge is adjacent to a sharp edge, its offset becomes a sheet model with a ruled surface instead of a tubular surface. If a disk is adjacent to a sharp edge, its offset is not generated. The other disks and wedges are offset in the same manner with the mathematical offset method. Although this modification breaks the mathematical definition of offsets, it cannot be avoided for the generation of acceptable thin-walled solid models.

The resulting offsets of the face-uses, wedges, and disks for the example model are shown in Figure 9. As shown in Figure 9(b), the offsets of the wedges on the sharp edges have planar surfaces, and the offsets of the wedges inside the sheet have tubular surfaces. As shown in Figure9(c), no offset model is generated for the disks on the sharp edges, and only the offset model of a disk inside the sheet is generated as a spherical patch.

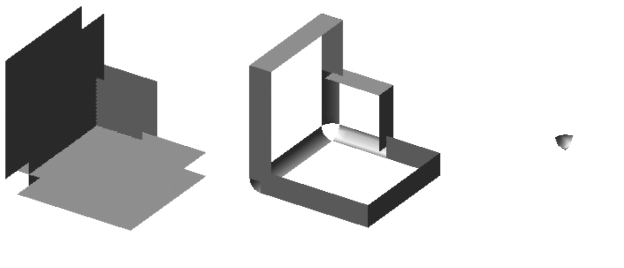


Figure 9. Offset models for the sample sheet model:
(a) partial-face offsets; (b) wedge offsets; (c) disk offsets

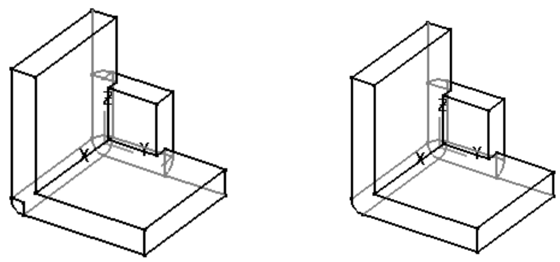
5.3. Union of the Face-Use, Wedge, and Disk Offsets

Once the offset models for the face-uses, wedges, and disks are generated, they are united into a single non-manifold model using the non-manifold Boolean operations. Most of the algorithms for non-manifold Boolean operations are based on the concept of *'merge and selection'* [12, 14, 15]. The merge and selection algorithm consists of two steps. In the first merge step, a non-manifold model, called a merge-set, is generated by uniting all primitives in the CSG tree of the Boolean operations. Then, in the next selection step, the topological entities that constitute the resulting body of the CSG tree are selected for its display or application. However, for our sheet thickening operations, the selection step is not necessary, as the union result is always what we want. Therefore, in the merge step of our algorithm, the system does not create any history records of topological entities, which are used in the latter selection step. The union operation of two non-manifold models, *A* and *B*, is composed of the following three steps.

(Step 1) Calculate the intersection points, curves, and surfaces between two models, *A* and *B*. This step is well known to be the most time consuming and error prone. To reduce the computing time and numerical errors, in our algorithm, the topological information of the original model is used for the union of the offset models made in Section 5.2. Each topological entity of an offset model has a pointer to its parent entity in the original sheet model. Before calculation of intersections of two entities, their parents are checked to see if they are identical. If so, the overlapping curves can be obtained easily by some additional checking of geometric properties of the two entities. Otherwise, the routine process for calculating intersections is used.

(Step 2) Generate new topological entities on the models *A* and *B* using the intersection points and curves. First, vertices are created at the intersection points and the end points of the intersection curves. Next, edges are created with the intersection curves. According to the intersection condition, one of the Euler operators is selected among MVL, MEV, SEMV, MEKL, and MEF. When new topological entities are created, the system also stores the partnership data, namely the pairs of topological entities of *A* and *B* that have the same geometry.

(Step 3) Copy all of the topological entities of *B* to *A*. The system finds out the *B*'s topological entities that do not have any partner in *A*, then copies them into *A* using Euler operators. The copy operations are carried out for the entities of low dimension then to higher ones, i.e., in order of the vertex, edge, and face. This method reduces the total number of Euler operators to be applied, as the boundary entities of higher dimensional entities are created in advance. The partnership data is referred to in order to find out the bounding vertices of an edge to be created, and the bounding edges of a face to be created. The union result of the offset models in Figure 9 is shown in Figure 10(a).



(a) (b)

Figure 10. Removing holes on thickness faces: (a) three holes on the thickness faces; (b) after removing holes

5.4. Filling Holes of Thickness Faces

When the union operations of the offset models are finished, a single unified non-manifold model is obtained. However, as no offset models are generated for the disks of the sharp edges, some holes may remain on the thickness faces. For instance, as shown in Figure 10(a), the united body of the offset models in Figure 9 has three holes on the thickness faces. In order to obtain a complete solid model shown in Figure 10(b), these holes should be detected and closed. To meet this requirement, we developed an algorithm for filling holes as automatically as possible. The procedure for filling holes is composed of two steps as follows.

(Step 1) Searching for the holes on the united offset model. In order to find the holes, sharp edges are first searched and stored in a list, and then the system checks if there is any loop composed of only sharp edges. Each loop is the boundary of a hole on the thickness faces. The sharp edges constituting a loop are extracted and used as input for generation of a new face at the hole. If any edge remains in the list after all sharp edges constituting loops are extracted, then such holes cannot be filled automatically. The user should fill the holes interactively using the modeling capabilities provided by our non-manifold modeling system.

(Step 2) Removing the holes searched. For the loops searched in Step 1, the holes are removed by applying MFKC or MFR operators with a plane or a freeform surface with an n-side boundary. Usually, if all of the sharp edges are included in *n* loops, MFKC is executed *n-1* times and then MFR is executed last.

If there are any sharp edges that do not constitute a loop in Step 1, then they need to be filled interactively by a user. Figure 11 shows an example for this case. Currently, the system highlights such sharp edges in red. Then, the user makes proper edges and fills the holes using the modeling capabilities of our non-manifold modeler.

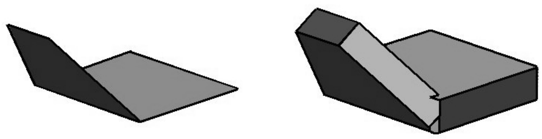


Figure 11. An exceptional case of removing holes on thickness faces

5.5. Removal of Unnecessary Topological Entities

When all holes of the united offset model are filled, the resulting model usually includes unnecessary entities such as wire edges and laminar faces, which are classified into nine cases as illustrated in Figure 12. Therefore, the removal operations for these entities should be carried out in order to obtain a complete solid model. In the mathematical offset algorithm proposed by Lee [7], these entities are found by checking if the distance to the original sheet model is less than the offset distance. However, in our practical offset algorithm, as the offset models of sharp edges have been generated with ruled surfaces instead of tubular surfaces, this criterion is no longer effective. In appreciation of the fact that the result is a solid model, we developed the purging operation: an operation that removes all unnecessary topological entities of a non-manifold model to convert it into a solid model. The algorithm for the purging operation is written in the C language style as shown in Figure 13.

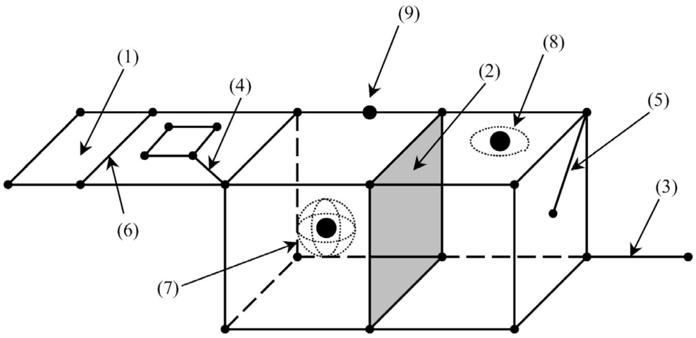


Figure 12. Topological entities to be purged: (1) laminar face; (2) screen face; (3) wire edge; (4) isthmus edge; (5) strut edge; (6) redundant edge; (7) single-vertex shell; (8) single-vertex loop; (9) redundant vertex

```
//
// Algorithm for the Purging Operation
//
for (each face of the model, f) {
    // (1) laminar face
    if  (f is adjacent to the same region in both sides)
        delete f with KFMC;
    // (2) screen face
    if  (f is adjacent to two different finite regions)
        delete f with KFR;
}
for (each edge of the model, e) {
    // (3) wire-edge
    if  (e is a wire)
      if (e is the only path connecting its two ending vertices)
            delete e with KEMS;
      else
            delete e with KEC;
    else
      if (e has two partial edges) {
          // (4) isthmus or (5) strut edge
          if (the partial edges belong to the same loop)
              delete e with KEML;
          // (6) redundant edge
          if (the faces of two partial edges have the
              same plane)
              delete e with KEF;
      }
}
for (each vertex of the model, v) {
    // (7) single-vertex shell
    if  (v is linked to a partial face)
        delete v with KVS;
    // (8) single-vertex loop
    else if (v is linked to a partial edge and has only one partial
        vertex)
        delete v with KVL;
    // (9) redundant vertex
    else if  (v is adjacent to two edges with the same geometry)
        delete v with JEKV;
}
```

Figure 13. Algorithm for the purging operation

6. OFFSETTING IN ONE-SIDE DIRECTION

The algorithm for sheet thickening in both the inside and outside directions has been described in Section 5 in detail. However, when plastic parts such as covers of electronics devices are designed with a feature-based solid modeler, the base feature for the main shape is generated first, and then the sub-features such as ribs and bosses are added to the base feature. In order to facilitate the base feature modeling, it is very desirable to develop a sheet thickening capability in one-side direction only. Let us assume that a given sheet is always two-manifold in the one-side sheet thickening algorithm, as the inside and outside are not clear in a non-manifold sheet model such as a T-shaped sheet. The one-side sheet thickening algorithm is the same as the both-side thickening algorithm, except for Steps 1 and 2. The algorithm is as follows.

(Step 1) Selection of topological entities to be offset:
The face-uses of a given sheet model can be grouped into the inside and the outside face-use groups. Only the face-uses on one side of the sheet are selected in this algorithm, whereas the face-uses of both sides are selected in the both-side sheet thickening algorithm. If the user selects a face-use, then the system searches for all face-uses of the group to be offset. In addition, the system searches for all wedges and disks on the face-uses to be offset selected. All sharp edges are searched as well.

(Step 2) Generation of offset models for the selected entities:
The offset models for the selected face-uses, wedges, and disks, except the wedges of the sharp edges, are the same as those of the both-side thickening algorithm. When generating a ruled surface for a thickness face, the profile line to be swept along the edge is half that of the both-side algorithm. The profile line can be defined as follows:

$$C(u) = E(0) + \mathrm{d}\, u\, \mathbf{N}_{FU}, \qquad 0 \le u \le 1$$

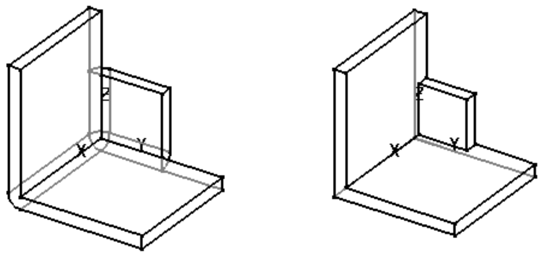
where $E(0)$ is the start position of the edge, d is the thickness, and $\mathbf{N}_{FU}$ is the unit normal of the face-use at $E(0)$. In addition, the original sheet model is copied and included in the offset models to be united in the next step, as this sheet model constitutes a wall of inside or outside. As mentioned previously, in order to save the computation time of intersections for Boolean operations, topological entities of the face-use, wedge, and disk offsets store their own parent entity. The parents of the entities of the copied sheet body are the corresponding entities of the original model.

(Step 3) Union of the face-use, wedge, and disk offsets:
Identical to Section 5.3

(Step 4) Filling holes of thickness faces:
Identical to Section 5.4

(Step 5) Removal of unnecessary topological entities:
Identical to Section 5.4

The example model in Figure 3 is thickened in the inside and outside directions, and the results are shown in Figure 14.



(a) (b)

Figure 14. One-side offset solids: (a) offset in the outside direction; (b) offset in the inside direction

7. CASE STUDY

Figure 15(a) shows a sheet model for the inside wall of the cover of a mouse device. By applying the sheet thickening operation in the outside direction, a solid model shown in Figure 15(b) is obtained.

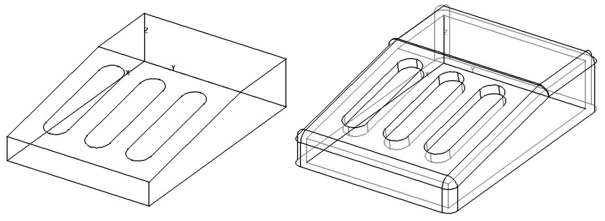


Figure 15. Modeling of a mouse: (a) a sheet model for the inner wall of a mouse; (b) thin-wall solid model obtained by offsetting in the outside direction

8. CONCLUSION

In this paper, we proposed sheet modeling and thickening methods based on a non-manifold topological representation for efficient solid modeling of thin-walled plastic or sheet metal parts. By adopting a non-manifold representation, the critical problems of the existing methods based on solid data structures can be solved as follows:

- All topological adjacency information can be interrogated from a topological database, whereas in the solid-based methods, geometric data needs to be evaluated to obtain such information.
- Any macros, for example the sheet Euler operators, do not need to be developed. As mentioned in the introduction, the standard Euler operators for solid models are still necessary for sheet modeling, as these macros are not a complete set of topological operators. However, the non-manifold Euler operators provide users with a complete set of topological operators for sheet modeling; that is, any sheet model can be modeled using only these operators.
- The algorithms for transformation of sheets into solids become simpler and more comprehensive by introducing and adapting general non-manifold offset algorithms.

However, in the current algorithms, all of the holes that lay on thickness faces cannot be removed automatically, and topological irregularity of an offset face caused by self-intersection is not yet considered. These aspects remain for future investigations.

REFERENCE

1. Lee, K. and Kwon, B.W., "Efficient Modeling Method of Sheet Objects", Computers in Engineering, *Proceedings of the International Computers in Engineering Conference and Exhibit* Vol.1, Published by ASME, New York, NY, USA, pp.437 - 446, 1992
2. Stroud, I., "Modeling with Degenerate Objects", *Computer Aided Design*, Vol.22, No.6, pp. 344-351, 1990
3. Lim, H.S. and Lee, K., "Efficient Solid Modeling via Sheet Modeling", *Computer Aided Design*, Vol. 27, No. 4, pp. 255-262, 1995
4. Lee, S.H. and Lee, K., "Feature-Based Non-manifold Geometric Modeling System to Provide Integrated Environment for Design and Analysis of Injection Molding Products", *Transactions of the Society of CAD/CAM Engineers*, Vol. 1, No. 2, 1996 (in Korean)
5. Lee, S.H. and Lee, K., "An Integrated CAD System for Mold Design in Injection Molding Process", Production Engineering Division, *The Winter Annual Meeting of the ASME*, Chicago, PED-Vol.32, pp.257-271, 1988
6. Forsyth, M., "Shelling and Offsetting Bodies", *Proceedings of Third Symposium on Solid Modeling and Applications*, Salt Lake City, Utah, USA, pp. 373-381, May 17-19, 1995.
7. Lee, S.H., "Offsetting Operations on Non-manifold Boundary Representation Models with Simple Geometry", *Proceedings of Fifth Symposium on Solid Modeling and Applications*, Ann Arbor, Michigan, pp. 42-53, 1999.6
8. Weiler, K., "Topological Structures for Geometric Modeling", *PHD. Thesis*, Rensselaer Polytechnic Institute, August 1986

9. Gursoz, E.L., Choi, Y., and Prinz, F.B., "Vertex-based Boundary Representation of Non-manifold Boundaries", Wozny, M.J., Turner, J.U., and Preiss, K., editors, *Geometric Modeling for Product Engineering*, North-Holland, pp. 107-130, 1990
10. Rossignac, J. and O'Conner, M.A., "SGC: A Dimensional-independent Model for Pointsets with Internal Structures and Incomplete Boundaries", *Geometric Modeling for Product Engineering*, North-Holland, pp. 145–180, 1990
11. Lee, S. H. and Lee, K., " Partial Entity Structure:A Compact Non-Manifold Boundary Representation Based on Partial Topological Entities", *Proceedings of Sixth Symposium on Solid Modeling and applications*, June 4-8, 2001 (to appear).
12. Masuda, H., "Topological operators and Boolean operations for complex-based non-manifold geometric models", *Computer Aided Design*, Vol. 25, No. 2, pp. 119-129, 1993.
13. Yamaguchi, Y. and Kimura, F., "Nonmanifold Topology Based on Coupling Entities", *IEEE Computer Graphics and Applications*, Vol.15, No.1, pp.42-50, January 1995
14. Crocker, G. A., and Reinke, W. F., "An Editable Non-manifold Boundary Representation", *IEEE Computer Graphics & Applications*, Vol. 11, No. 2, pp. 39-51, March 1991.
15. Gursoz, E. L., Choi, Y. and Prinz, F. B., "Boolean Set Operations on Non-manifold Boundary Representation Objects", *Computer Aided Design*, Vol. 23, No. 1, pp. 33-39, January/February 1991.