Offsetting Operations on Non-manifold Boundary Representation Models with Simple Geometry

Sang Hun Lee

School of Mechanical and Automotive Engineering,
Kookmin University

ABSTRACT

This paper describes non-manifold offsetting operations that add or remove a uniform thickness from a given non-manifold object with simple geometry. Each offsetting operation for wireframes, sheets and solids is applicable to different engineering areas with a great potential usefulness. However, the representation schemes of conventional geometric modeling systems have not described all of the wireframes, sheets and solids together; each offsetting capability has been developed and applied separately in each system. In recent years, non-manifold geometric modelers have been developed and more widely spread. Since they can manipulate different levels of models with a unified data structure, these three types of offsetting operations can be integrated into one. Moreover, non-manifold offsetting operations can be used to give flesh to abstract models that are generated as a mixture of wireframes and sheets in conceptual design. Therefore, in this paper, the mathematical definitions and properties of the non-manifold offsetting operations are described first and then an offset algorithm using the non-manifold Euler and Boolean operations is suggested. In this algorithm, offset models for all or a subset of the vertices, edges and faces of a given non-manifold model are generated first. Then, they are united into one body using the non-manifold Boolean operations. Finally, all topological entities that are within offset distance are detected and removed in turn. In addition to the general offset algorithm, this paper discusses its variations for wireframes and sheets to provide the more practical offset solids for pipelines, plastic parts and sheet metal parts.

Key words: CAD, algorithms, geometric modeling, offset, non-manifold, wireframe, sheet, solid

861-1, Jeongneung-Dong, Seongbuk-Gu, Seoul 136-702, Korea
E-mail: shlee@kmu.kookmin.ac.kr

Fifth Symposium on Solid Modeling Ann Arbor MI

1 INTRODUCTION

1.1 Background and Objective

Offsetting operations add or remove a uniform thickness from a given object. To offset an object X by a positive distance r is to add to X all the points exterior to X that lie within a distance r from the boundary of X. For a negative offset, one subtracts from X all the points of X within a distance r from its boundary. Offsetting operations can be applied to not only solid models but also sheet or wireframe models. Here, a sheet model is defined as a degenerated solid model with zero thickness, and thus it looks like a generalized surface model that allows an edge to be adjacent to more than two faces. Potential applications of offsetting operations cover a wide range. Wireframe offsetting can be used to generate sheet or solid models for pipelines of plants or ships. Sheet offsetting has been used to generate solid models for plastic or sheet metal parts with thin and uniform thickness efficiently [8, 10, 18]. Solid offsetting has been used for tolerance analysis, clearance testing, NC tool path generation, planning collision-free paths for robots, constant-radius rounding and filleting and so on [15].

Since conventional geometric modeling systems usually do not represent all of the wireframe, surface and solid models with a single representation scheme, offsetting operations for each type of model have to be developed separately and used in their own restricted application areas. In recent years, however, non-manifold geometric modeling systems, which can manipulate all of them with a single unified topological representation, have been developed and more widely spread. Therefore, if we develop offsetting operations on non-manifold models, three types of operations can be integrated into one in a single environment and used in many application areas. Moreover, these operations can be used in detailed design to create a solid from an abstract model that is usually generated as a mixture of wireframes and sheets in conceptual design [6].

To meet this requirement, algorithms for non-manifold offsetting operations are proposed and implemented using the non-manifold Euler and Boolean operations. The remainder of this paper is organized as follows: The rest of this section surveys the related work. Section 2 describes the formal definitions and mathematical properties of non-manifold offsetting. Section 3 introduces a non-manifold geometric modeling system in which non-manifold offsetting operations are implemented. Section 4 describes a non-manifold offset algorithm in detail. It uses the modeling capabilities of a non-manifold modeler like Euler and Boolean operations. Section 5 discusses variational offset algorithms that allow users to create more practical offset solids from wireframes and sheets. Section 6 shows a case study.

Finally, Section 7 presents the conclusion of this paper.

1.2 Related Work

A lot of research related with offsetting has been carried out for over three hundred years and it may be classified into two main categories: offset geometry and offset topology. The area of offset geometry deals with the exact or approximate methods for generating offset curves and surfaces, which are well surveyed in Pham's paper [14]. The area of offset topology deals with the development of topological operations for generating offset solids or converting sheets into solids in geometric modeling systems. This paper belongs to the research category of offset topology because it is concentrated in the description of topological operations for offsetting a non-manifold model. Although no publications on three-dimensional non-manifold offsetting were found in journals, much research on offsetting operations on solids and sheets has been made and published. Since offsetting operations on non-manifold objects encompass those on solids, sheets and wireframes, previous works on solid and sheet offsetting will be reviewed as related work instead.

Rossignac and Requicha [15] first attempted to incorporate offsetting operations into solid modelers. They introduced a family of operations called solid offsetting (s-offsetting). The regularized solid offset of a regular set S by a distance r can be seen as the volume swept by a solid sphere of radius r as its center moves throughout the set S. They discussed some properties of s-offsetting operations, and included them in an extended form of CSG called CSGO, in which offsetting operations are represented as non-terminal nodes in CSG trees. This method allows CSG modelers to get around the problem that, if a solid is some Boolean combination of primitives, the offset operator cannot be expressed as the same Boolean combination of the offsets of the primitives.

Farouki [2] proposed an offsetting procedure for simple solid primitives of extrusion and of convex polyhedrals. Offset patches for each face, edge and vertex are generated first, and then they are merged together to compose the boundary of an offset solid. Farouki described exact offset procedures for simple solids of three types: convex planar polyhedra, solids of revolution and solids of linear extrusion with simple profile curves. The simple profile curves are linear and circular arcs. In order to construct the offset to a simple solid, its topology is first resolved as faces, edges and vertices. For each face, a face offset element is generated by offsetting the face by a vector r **n**, where **n** is the outward normal to that face, and r is the offset distance. For each edge, edge offset element is generated by a translation sweep of an arc along the edge curve. For each vertex, a spherical vertex offset element is generated as a set of triangular spherical patches bounded by circular arcs. The face, edge and vertex offset elements above are guaranteed to match precisely at their boundaries and form a completed designed offset surface for the simple solids. However, this algorithm cannot be applied to a solid with concave edges or vertices or a solid with complex curves and surfaces.

Saeed, *et al.* [16] also attempted to introduce a class of offsetting operations into solid modeling systems. Their mathematical formulation for offsets is based on the concept of open ball neighborhoods in an n-dimensional space. The definition of an offset is equivalent to that given by Rossignac and Requicha [15]. By using the neighborhood function, an offset solid can be constructed from the offsets of its boundary sets in lower dimensions. This mathematical formulation provides a coherent framework for synthesizing an offset solid.

Satoh and Chiyokura [17] defined the open set to represent the partial boundary of a solid, and developed an algorithm for Boolean operations on open sets. As an application of the open-set Boolean operations, they also proposed algorithms for offset solid generation and self-intersecting solid correction. Their offset procedure is composed of the following steps. First, open sets with offset surfaces are generated for all faces of a given solid. Next, they are united using the open-set Boolean operations. Finally, if there are any gaps between the open sets, new faces are generated to close the gaps. If the resulting offset solid is self-intersecting, the correction procedure is applied. However, they did not describe a detailed method of eliminating gaps that are caused by convex edges or vertices.

Forsyth [3] proposed algorithms for offsetting and shelling operations on B-rep solids, and implemented them with the modeling capabilities of SolidDesigner. In his offset algorithm, an offset solid is generated in the following manner. First, offset surfaces for each face are generated. Secondly, offset curves for each edge are generated by intersecting two offset surfaces of the adjacent faces. Thirdly, offset positions for each vertex are also obtained by intersecting offset curves. Fourthly, the offset surfaces, curves and positions are attached to the corresponding faces, edges and vertices, respectively. Finally, if positive offsetting, all convex edges are blended with a radius of a given offset distance r, or, if negative offsetting, all concave edges are blended. The outstanding feature of this algorithm is to substitute the geometric entities of the given solid with the offset ones and then blend the concave or convex edges. However, the offset solid can be topologically irregular in geometric substitution, but he did not suggest any topology correction procedures.

In addition to solid offsetting, research on sheet offsetting has been done in order to more efficiently generate solid models of thin and constant thickness from a given sheet model. This function is very useful in modeling plastic or sheet metal parts. In most existing algorithms, a sheet object is modeled as a degenerate solid model with zero thickness, and then it is converted to a solid by replacing its geometry with the offset one. Stroud [19] 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 the so-called 'sharp' edges. In the transformation procedure, the sharp edges and their vertices are first split to form the boundary edges of the thickness faces. Then, the geometry for each vertex, edge and face are calculated and set to the corresponding topological entities. However, this transformation method may result in practically unacceptable solids, but he did not suggest any correction methods for them.

Lee and Kwon [8] proposed another sheet modeler and transformation procedure, which was revised by Lim and Lee [11] afterwards. In this system, Winged Edge Data Structure is used to represent sheet objects. It is similar to Stroud's from the viewpoint of representation scheme because both of them use solid data structures in order to represent sheet objects. However, in Lee and Kwon's system, a sheet model has full topological data for the corresponding solid model including the thickness faces. Thus, when a sheet is transformed into a solid, it is assumed that only the geometry is replaced with the offset one. However, this assumption may cause practically unacceptable solid models. Therefore, they also had to suggest a method of correcting such unacceptable solids. In addition, in order to make it easier to develop sheet modeling functions, they proposed a set of sheet Euler operators that are like macros of the standard Euler operators

for solid modeling. However, since this is not a complete set of topological operators for generating and deleting topological entities of sheet models, the standard Euler operators have to be used together.

Stroud and Lee's methods have a common problem that the topological representation does not have proper information about the adjacency relationships between the topological entities. This is because they describe sheet objects, which is originally non-manifold, with B-rep solid data structures. This defect makes it difficult to develop the modeling capabilities for sheets as well as the transformation procedures of sheets into solids. To solve this problem, Lee and Lee [10] introduced a non-manifold B-rep as a framework of a unified sheet/solid modeler. They also suggested an algorithm for sheet conversion into a solid in a non-manifold modeling environment. In this algorithm, offset patches are first generated as non-manifold sheet models for all face-uses and all convex edges and vertices. Next, they are united into one, using non-manifold Boolean operations. Then, the faces, which are within a given offset distance r from the original sheet model, are removed in turn. Finally, the thickness faces are generated using the non-manifold Euler operators. Although the step for the generation of thickness faces needs to be enforced, this algorithm is more comprehensive and simpler than the previous methods.

2 MATHEMATICAL DEFINITIONS AND PROPERTIES OF NON-MANIFOLD OFFSETS

2.1 Definitions of Non-manifold Models

In this paper, the Euclidean cell complex is selected as a proper mathematical model for a non-manifold object. In an n-dimensional Euclidean space, E^n, the n-dimensional cell (n-cell) is defined as a bounded subset of E^n that is homeomorphic to an n-dimensional open sphere. If a set of a finite number of cells, X, satisfies the three conditions below, X is defined as an Euclidean cell complex [12].

$$X = \cup_{\lambda \in \Lambda} e_\lambda , \tag{1}$$

$$[e_\lambda] - e_\lambda \subset \{ e_\mu | \dim(e_\mu) < \dim(e_\lambda) \}, \quad \mu \in \Lambda, \lambda \in \Lambda , \tag{2}$$

$$e_\lambda \cap e_\mu = \phi, \quad \lambda \neq \mu, \ \mu \in \Lambda, \ \lambda \in \Lambda, \tag{3}$$

where e_λ denotes an n-cell, $dim(e_\lambda)$ the dimension of e_λ, and $[e_\lambda]$ the closure of e_λ that includes an n-cell as well as its boundary. The first condition means that an n-dimensional cell complex is a collection of 0-cells, 1-cells, 2-cells, ..., and n-cells. The second condition means that the boundary of each cell consists of lower dimensional cells. This condition ensures that a cell complex is always closed and does not contain any unclosed topological entities. The third condition means that no topological entities intersect each other.

In this paper, since we deal only with non-manifold objects in E^3, their models are always three-dimensional cell complexes. The non-manifold modeler adopted in this paper supports the modeling capabilities for these cell complexes. Its brief description is given in the following section. When comparing n-cells with topological entities of non-manifold modelers, 0-cells, 1-cells, 2-cells and 3-cells correspond to vertices, edges, faces and regions, respectively. In our non-manifold modeler, all three-dimensional spaces are represented by the region entities, whether or not they are filled with material. The material information is stored as an attribute for a region. In this paper, an empty region is called a *void* region and a material-filled region is a *filled* region.

2.2 Definitions and Properties of Non-manifold Offsets

(1) Definition of Non-manifold Offsets

If X denotes a non-manifold model defined as a 3-D cell complex and $X \oplus r$ denotes the positive offset of X by a positive distance r, then the positive offset of X, X_o, is

$$X_o = X \oplus r = \{ \mathbf{p}_o | \ \exists \mathbf{p} \in X, \| \mathbf{p}_o - \mathbf{p} \| \leq r \} . \tag{4}$$

Note that, if X is empty, X_o is also empty. An equivalent definition is as follows:

$$X \oplus r = \cup_{\mathbf{p} \in X} B^*(\mathbf{p}, r) \tag{5}$$

where $B^*(\mathbf{p}, r) = \{ \mathbf{p}_o | \ \| \mathbf{p}_o - \mathbf{p} \| \leq r \}$ and denotes a closed ball of radius r centered at $\mathbf{p}$. This can be understood as the volume swept by a solid sphere of radius r as its center moves throughout X. This definition is equivalent to the Minkowski sum of a solid sphere and the original model X [5, 13].

The complement of a non-manifold model can be obtained easily by exchanging the *void* and *filled* attributes of the regions with each other. If the negative offset of X by a distance r is denoted by $X \ominus r$ and the complement of X is denoted by X^C, the negative offset is defined as follows:

$$X \ominus r = (X^C \oplus r)^C \tag{6}$$

Fig. 1(a) shows a cross-section of a simple non-manifold model that is composed of an L-shaped solid and a sheet. Its positive offset is shown in Fig. 1(b) and its negative offset in Fig. 1(c).

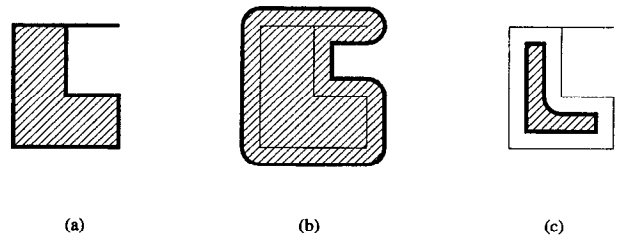


Fig. 1 Non-manifold offsetting operations (a) a simple non-manifold object composed of a sheet and a L-shaped solid (b) a positive offset (c) a negative offset

(2) Mathematical Properties of Non-manifold Offsets

Various mathematical properties of solid offsets were discussed by Rossignac and Requicha [15]. Most of them also appear in offsets of non-manifold models. The equality and inclusion relations between solid offsets also appear in non-manifold offsets. That is, if $A = B$, then $A \oplus r = B \oplus r$ and $A \ominus r = B \ominus r$. In general, however, neither $A \oplus r = B \oplus r$ nor $A \ominus r = B \ominus r$ implies that $A = B$. If $A \subset B$, then $A \oplus r \subset B \oplus r$ and $A \ominus r \subset B \ominus r$.

In addition, non-manifold models defined as a 3-D Euclidean cell complex are algebraically closed under offsetting operations. That is, if X is a 3-D Euclidean cell complex, then its offsets $X \oplus r$ and $X \ominus r$ are also 3-D Euclidean cell complexes. This implies that one can add offsetting operations to a non-manifold modeler and be sure that the resulting sets are valid models and therefore can be used in the system as inputs for further operations.

However, a positive offsetting operation does not generally commute with a negative offsetting operation, and the two

operations should not be thought of as inverses because they have the following properties:

$$(X \ominus r) \oplus r \subset X \subset (X \oplus r) \ominus r \qquad (7)$$

Actually, we can obtain the rounding and filleting effects of non-manifold models by combining positive and negative offsetting operations. That is, $(X \ominus r) \oplus r$ rounds the convex edges and vertices of the given object, while $(X \oplus r) \ominus r$ fillets the concave edges and vertices [15].

The topological boundary of an expanded model $X \oplus r$ or a shrunk model $X \ominus r$ is included in the set of points that are at a distance r from X. If $d(\mathbf{p}, X)$ denotes the distance from a point $\mathbf{p}$ to the closest point on X, this relationship can be written as follows:

$$\partial(X \oplus r) = \{ \mathbf{p}_o \mid d(\mathbf{p}_o, X) = r \} \qquad (8)$$

$$\partial(X \ominus r) = \{ \mathbf{p}_o \mid d(\mathbf{p}_o, X^C) = r \}, \qquad (9)$$

where

$$d(\mathbf{p}_o, X) = min \| \mathbf{p}_o - \mathbf{p} \|, \quad \mathbf{p} \in X \qquad (10)$$

and

$$d(\mathbf{p}_o, X) = d(\mathbf{p}_o, \partial X) \qquad (11)$$

The boundary of the offset model has the following property as in a solid offset :

$$\partial(X \oplus r) \subset \partial(\partial X \oplus r) \qquad (12)$$

This property is useful for constructing a superset of the boundary of the offset model. $\partial X \oplus r$ can be obtained by uniting the offsets of all faces, edges and vertices of the given model X. Since the boundary of the resulting offset model $\partial(X \oplus r)$ is included in $\partial(\partial X \oplus r)$, it can be obtained by eliminating unnecessary topological entities from $\partial(\partial X \oplus r)$. Based on these mathematical definitions and properties, an offset algorithm for a non-manifold model is suggested in Section 4.

3 NON-MANIFOLD GEOMETRIC MODELING SYSTEM

3.1 Data Structure

In this paper, ANYSHAPE is used as a kernel engine for non-manifold geometric modeling, which has been developed by the author based on Partial Entity Data Structure (PEDS) and object-oriented programming technology [9]. The topological entities of PEDS are classified into two groups: One is the primary entities that appear commonly in most existing topological representations. They are vertices, edges, faces, regions, loops and shells. The other is the secondary entities that are introduced to represent the non-manifold adjacency relationships between the primary entities. As shown in Fig. 2, ANYSHAPE has the partial faces (p-face), the partial edges (p-edge) and the partial vertices (p-vertex) as the secondary entities.

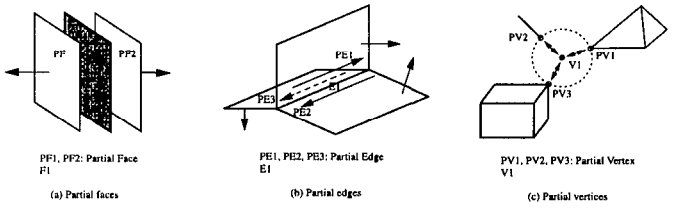


Fig. 2 Partial topological entities

A partial face is a side of a face and is a part of the shell that is the boundary of a region. It was introduced to represent a cellular model with interior partitions. A face is always associated with two partial faces (*mate* relation) on either sides of it. The partial edge is a use of an edge in a face. It was introduced to represent the non-manifold condition that an edge is adjacent to more than two faces. The number of partial edges around an edge is the same as the number of faces around an edge. They are ordered in loops and edges. The partial vertex was introduced to represent the non-manifold condition that a vertex is adjacent to more than two *2*-manifold surfaces. The number of partial vertices associated with a vertex is the same as the number of *2*-manifold surfaces.

The schematic diagram of PEDS is shown in Fig. 3 and is a familiar hierarchical data structure. More detailed descriptions on this data structure and the comparison with other schemes are given in Lee and Lee's work [9].

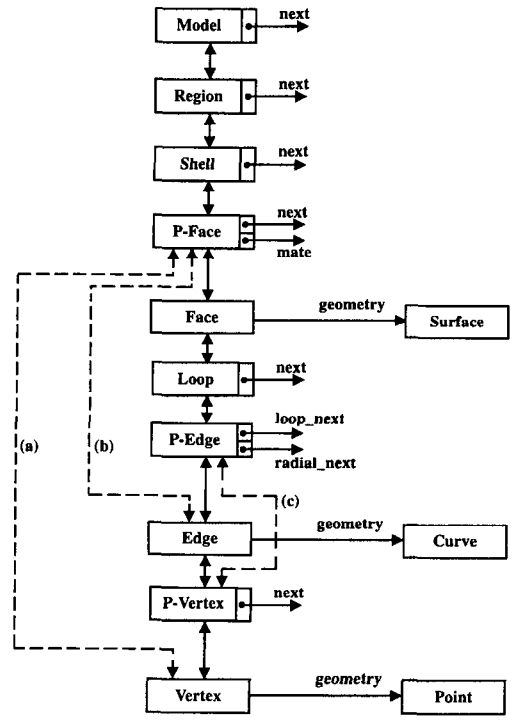


Fig. 3 Partial Entity Data Structure

ANYSHAPE has different levels of geometric modeling capabilities like Euler operators, primitives, sweeping operations, Boolean operations and so on. The offsetting operations are developed using these capabilities and are implanted into ANYSHAPE.

3.2 Euler operators

Most existing B-rep solid modelers use Euler operators to maintain the topological integrity of the models in implementing high-level modeling operations. Several researchers such as Yamaguchi [20], Masuda [12] and Lee [9] have derived proper Euler formulas that can be applied to non-manifold models and proposed a set of Euler operators which is applicable to the corresponding topological structure. The Euler formula and

Euler operators used in ANYSHAPE are the ones proposed by Lee and their brief description is given in the following paragraphs.

The basic formula that can be applied to our topological data structure is as follows:

$$V - E + F - L = S - C + R, \qquad (13)$$

where V, E and F are the number of vertices, edges and faces, respectively, L is the number of hole loops, S is the number of void shells in the regions, C is the number of cycles which cannot be contractible to a point, and R is the number of regions. From Eq. (13), we can see that six independent Euler operators and their six inverse operators are sufficient to manipulate the topological structure of non-manifold models in our system. For the convenience of the modeling operations, however, eight additional Euler operators are supplemented, and two topological operators not directly related to the Euler formula are added that generate or delete a model and a dummy infinite region when initially creating a model or finally deleting it. The basic and extended Euler operators in ANYSHAPE are listed in Table 1 and illustrated in Fig. 4. All of the high-level modeling operations, such as sweeping and Boolean operations, are implemented using the twenty Euler operators in our kernel modeler.

Table 1 Euler operators

Class	Name	Description
Basic Euler Operators	MEV(KEV) MEC(KEC) MFKC(KFMC) MFR(KFR) MVS(KVS) MVL(KVL)	make(kill) edge, vertex make(kill) edge, cycle make(kill) face, kill(make) cycle make(kill) face, region make(kill) vertex, shell make(kill) vertex, loop
Extended Euler Operators	SEMV(JEKV) MEF(KEF) KEML(MEKL) KEMS(MEKS)	split(join) edge, make(kill) vertex make(kill) edge, face kill(make) edge, make(kill) loop kill(make) edge, make(kill) shell
Topological Operators	MMR(KMR)	make(kill) model, region

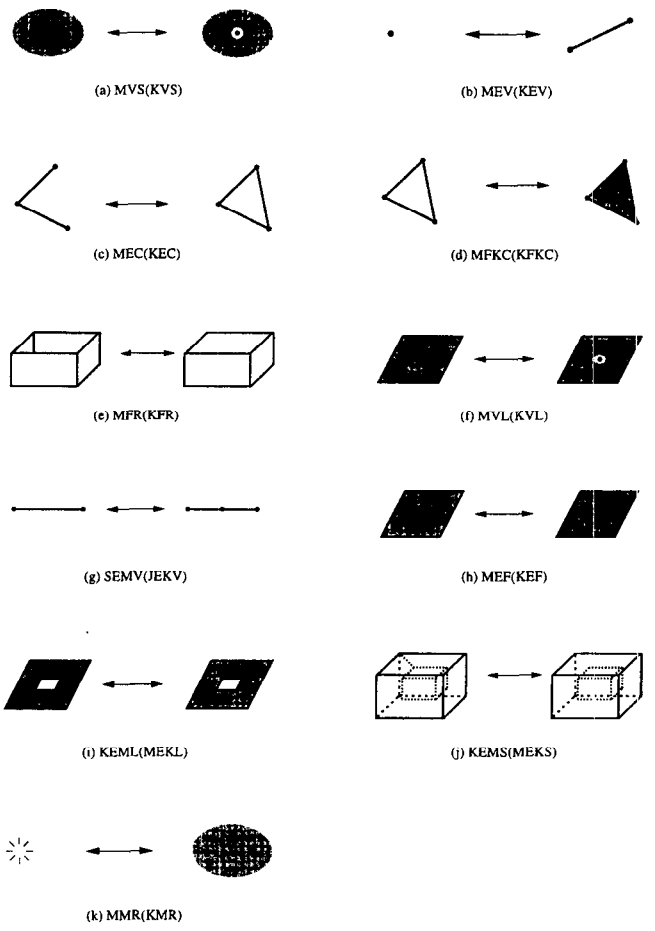


Fig. 4 Non-manifold Euler Operators

4 NON-MANIFOLD OFFSET ALGORITHM

This section presents an algorithm for non-manifold offsetting operations, which is based on the mathematical definitions and properties of non-manifold offsets described in Section 2. The algorithm focuses on the positive offset of a given non-manifold model X, because the negative offset is easily obtained by a sequential process of Eq. (6). The offset algorithm for non-manifold models can be summarized as follows:

1. When a negative offset is desired, execute the complementing operation on the given model first of all. The complement of X can be easily obtained simply by replacing the region attribute *void* with *filled* and vice versa.
2. Select all the vertices, edges and faces whose offsets will contribute to the construction of a superset of the boundary of the offset model of X.
3. Generate the offset sheet/solid models for the vertices selected in Step 2.
4. Generate the offset sheet models for the edges selected in Step 2.
5. Generate the offset sheet models for the faces selected in Step 2.
6. Unite all the offset models generated in Step 3 to 5 in order to create a superset model X_s whose boundary is $\partial(\partial X \oplus r)$.
7. Remove all topological entities of X_s that are within the offset distance r from the boundary of the original model X to obtain the exact offset model X_o.
8. When negative offsetting, take the complement of X_o.

The steps from 2 to 7 are described in the sections 4.1 to 4.6 in detail.

4.1 Selection of Topological Entities to Be Offset

The system searches for all the shells of the void regions of X and then selects all the faces, edges and vertices that are adjacent to the shells. The offsets of these entities will participate in the boundary of the offset model of X. In the case of a model shown in Fig. 5, since the shells S_0, S_1 and S_3 are the boundaries of the void regions R_0 and R_2, the collected entities are all the faces, edges and vertices adjacent to S_0, S_1 and S_3. F_1 is excluded because it is adjacent to only S_2 that is a shell of the filled region R_1. In Partial Entity Data Structure, shells are oriented to the inner space of regions. Each of the selected entities is offset towards the inner normal directions of its adjacent void shells.

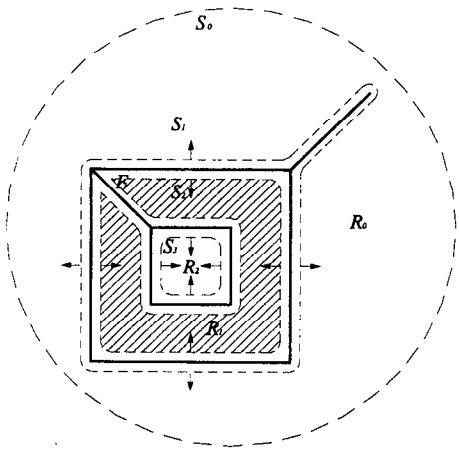


Fig. 5 A simple non-manifold model

4.2 Generation of Vertex Offsets

Offset models are generated for the vertices selected in Section 4.1. The positive offset of a vertex by a distance r is defined as a sphere of radius r centered at the vertex position. However, it is not necessary to generate spheres for all the vertices. Spheres need to be generated only for the vertices of singular points on the boundary of X. In addition, spheres do not need to be a whole one because, in some cases, only partial segments of the spheres are sufficient for vertex offsets. In this paper, the vertices are classified into several types as shown in Table 2. In cases 1.1 and 3.1, a whole sphere is generated as a vertex offset using primitive generation capabilities in ANYSHAPE. However, in other cases, the vertex offsets are only spherical segments or do not need to be generated. Detailed modeling procedures are described in the following subsections.

Table 2 Positive vertex offsets

Classification	Example	Condition	Positive offset result
1. Single-vertex shell	○	Always	Sphere
2. Wire-edge vertex		If two edges join smoothly (G^1 cont)	Ø
		In the case of an end vertex	Sphere (hemi)
		Otherwise	Sphere (partial)
3. Single-vertex loop		In the case of a singular point	Sphere (partial)
		Otherwise	Ø
4. Sharp-edge vertex		If two edges join smoothly	Ø
		Otherwise	Sphere (partial)
5. Inner vertex		If smooth	Ø
		If concave	Ø
		Otherwise	Sphere (partial)

(1) Vertices on the Tip of a Wire Edge (Case 2.2)

If the vertex is on the tip of a wire edge as shown in Fig. 6, only the sheet model of a hemisphere is enough for the vertex offset. The boundary of the hemisphere is a circle and its equation can be described as follows:

$$C(t) = \mathbf{p}_v + r\,(\,cos(t)\,\mathbf{n} + sin(t)\,\mathbf{b}\,),\ 0 \le t \le 2\pi \qquad (14)$$

where $\mathbf{p}_v$ is the vertex position, $\mathbf{n}$ is a unit normal to the edge curve, and $\mathbf{b}$ is a unit binormal that is calculated by $\mathbf{b} = \mathbf{t} \times \mathbf{n}$ if $\mathbf{t}$ is the unit tangent of the edge curve at $\mathbf{p}_v$. In ANYSHAPE, as shown in Fig. 6, a hemisphere is usually represented by four triangular spherical segments for easy topology manipulation. The algorithm to generate a hemisphere for the vertex offset is composed of four sequential steps as follows:

1. Calculate four points that quadrisect the edge curve $C(t)$, i.e., $C(0)$, $C(\pi/2)$, $C(\pi)$ and $C(3\pi/2)$, and the top point $\mathbf{p}$ of the hemisphere with the equation $\mathbf{p} = \mathbf{p}_v + r\,\mathbf{t}$.
2. Generate a wireframe model of the circle using the following sequence of Euler operators:

   MMR → MVS at $C(0)$ → MEV from $C(0)$ to $C(\pi/2)$ → MEV from $C(\pi/2)$ to $C(\pi)$ → MEC from $C(3\pi/2)$ to $C(0)$
3. Generate four longitudinal edges by connecting the top point $\mathbf{p}$ of the hemisphere and four vertices on the circle as follows. The resulting body is a wireframe like a birdcage.

   MEV from $C(0)$ to $\mathbf{p}$ → MEC from $\mathbf{p}$ to $C(\pi/2)$ → MEC from $\mathbf{p}$ to $C(\pi)$ → MEC from $\mathbf{p}$ to $C(3\pi/2)$.
4. Attach four spherical faces to the cage windows by calling MFKC four times.

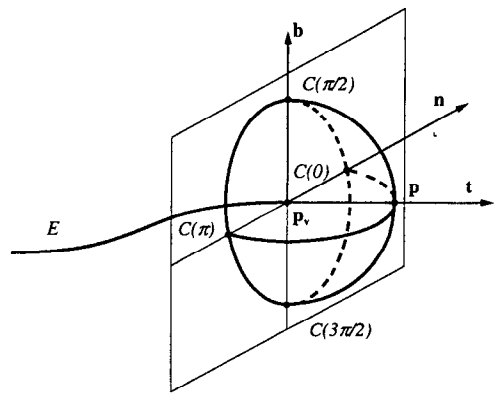


Fig. 6 Creating a hemisphere for the vertex offset

(2) Singular Vertices Adjacent to Only Two Edges (Case 2.3 and 4.2)

If a vertex is adjacent to only two edges of a wireframe or a sheet and does not satisfy G^1 continuity as shown in Fig. 7, a spherical segment, not a whole sphere, is enough for the positive vertex offset. In this case, a sheet model of the spherical segment is generated by creating a semicircle as the profile and performing a rotational sweep about the axis $\mathbf{n}_1$ by the angle υ. Here, υ denotes the angle between two edge tangents $\mathbf{t}_1$ and $\mathbf{t}_2$ at the vertex position, and $\mathbf{n}_1$ is a unit normal obtained by $\mathbf{n}_1 = \mathbf{t}_1 \times \mathbf{t}_2$. The curve equation of the semicircle C_l for E_l is as follows:

$$C_l(t) = \mathbf{p}_v + r\,(\,cos(t)\,\mathbf{n}_1 + sin(t)\,\mathbf{b}_1\,),\qquad 0 \le t \le \upsilon \qquad (15)$$

where $\mathbf{p}_v$ is the vertex position and $\mathbf{b}_1$ is a unit vector obtained by $\mathbf{b}_1 = -\mathbf{t}_2$.

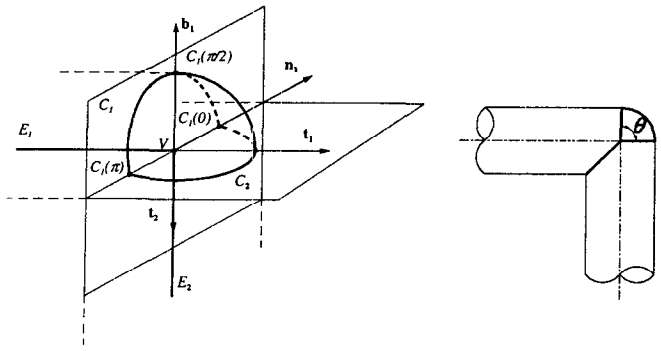


Fig. 7 Creating a spherical patch for a vertex offset

The algorithm to generate the vertex offset for a singular vertex adjacent to only two edges can be described as follows:

1. Calculate the start, end and bisectional points of C_l, i.e., $C_l(0)$, $C_l(\pi/2)$ and $C_l(\pi)$, using Eq. (15).
2. Generate a wireframe model for the semicircle C_l. In ANYSHAPE, a semicircle is usually constructed by two edges of quadrant for easy calculation of geometric properties. A sequence of Euler operators is applied as follows:

   MMR → MVS at $C_l(0)$ → MEV from $C_l(0)$ to $C_l(\pi/2)$ → MEV from $C_l(\pi/2)$ to $C_l(\pi)$
3. Generate a sheet model through the rotational sweeping of the semicircle by the angle υ.

(3) Convex Singular Vertices on a 2-Manifold Surface (Case 5.3)

When vertices are on two-manifold surfaces, they can be classified into convex, concave, smooth and saddle surfaces. If a vertex is convex, a sphere is generated as the vertex offset by default, otherwise nothing is generated in this system. In the case of a convex vertex, the system generates a spherical segment only if all the edges adjacent to the vertex are convex.

For example, when a convex vertex is adjacent to three edges as shown in Fig. 8, a spherical segment is created. Its boundary is composed of three circular arcs of radius r centered at the vertex position $\mathbf{p}_V$. The start and end points are easily calculated by the following equation:

$$\mathbf{p}_{Vi} = \mathbf{p}_V + r\,\mathbf{n}_{Fi}, \quad i = 1,2,3 \qquad (16)$$

where $\mathbf{p}_{Vi}$ are intersection points between three arcs, $\mathbf{p}_V$ is the vertex position, and $\mathbf{n}_{Fi}$ is a face normal at $\mathbf{p}_V$. The unit tangents $\mathbf{t}_i$ for each edge become the normals of planes on which circular arcs lie. In order to define each circular arc, a formula similar to Eq. (15) is used here again. Regarding the arc for E_l, the unit tangent $\mathbf{t}_1$ at $\mathbf{p}_V$ becomes the normal of the plane in which the arc lies if $\mathbf{p}_V$ is the start point of E_l. If C_l denotes the parametric circular arc for E_l and θ is the angle from $\mathbf{n}_1$ to $\mathbf{n}_{F1}$, C_l is obtained by the following equations:

$$C_l(t) = \mathbf{p}_v + r\,(\,cos(t)\,\mathbf{n}_{F1} + sin(t)\,\mathbf{b}_1\,), \qquad 0 \le t \le \theta \quad (17)$$

where

$$\mathbf{b}_1 = \mathbf{t}_1 \times \mathbf{n}_{F1} \qquad (18)$$

The other arcs for two remaining edges can be obtained in the same manner.

Once all geometric data are prepared, a sheet model for a spherical segment is generated using a proper sequence of Euler operators. In the case of Fig. 8, the sequence is as follows:

MMR → MVS at $\mathbf{p}_{V1}$ → MEV from $\mathbf{p}_{V1}$ to $\mathbf{p}_{V2}$ → MEV from $\mathbf{p}_{V2}$ to $\mathbf{p}_{V3}$ → MEC from $\mathbf{p}_{V3}$ to $\mathbf{p}_{V1}$ → MFKC

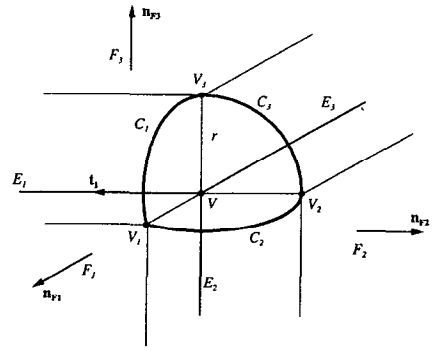


Fig. 8 Creating a spherical patch for a vertex offset

4.3 Generation of Edge Offsets

The boundary of the positive edge offset is a set of points that are at a distance r from the edge. It also can be viewed as a rolling ball surface of radius r when its center moves along the edge curve as a trajectory. As shown in Fig. 9, the rolling ball surface can be divided into three segments; two spherical segments S_1 and S_2 centered at the two ending points, and a tubular surface S_3 along the edge curve. Since two spherical segments were already generated in the vertex offset procedure at Section 4.2, only a tubular segment is generated with a sheet model for the edge offset.

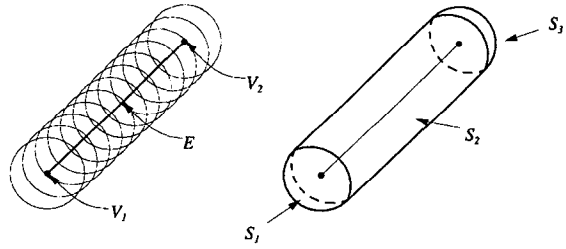


Fig. 9 Offsetting an edge by rolling a ball along the edge

The tubular surface can be viewed as a surface swept by a circle of radius r as its center moves along the edge curve. As illustrated in Fig. 10, if the edge curve is denoted by $E(u)$, the tubular surface S is as follows:

$$S(u,v) = E(u) + r\,(\,cos(v)\,\mathbf{n}(u) + sin(v)\,\mathbf{b}(u)\,) \qquad (19)$$

where $\mathbf{t}(u)$ is a unit tangent of $E(u)$, $\mathbf{n}(u)$ is a unit normal, and $\mathbf{b}(u)$ is a unit binormal that is calculated by $\mathbf{b}(u) = \mathbf{t}(u) \times \mathbf{n}(u)$.

A tubular surface can be self-intersecting. However, the surface does not self-intersect, if it has the following property:

$$r < min|1/\kappa(u)|, \qquad (20)$$

where $\kappa(u)$ means the curvature of $E(u)$. To manipulate self-intersecting surfaces is very difficult and is one of the main research issues in the offset geometry area. However, since this paper is focused on the topology construction of

offset models, it is assumed that all of the offset surfaces satisfy the property of Eq. (20) and it remains as future work to extend this algorithm to encompass self-intersecting offset surfaces.

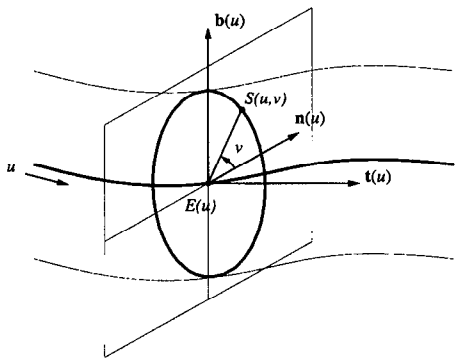


Fig. 10 Creating a tubular surface for an edge offset

Note that not all the edge offsets may appear in the resulting offset model and, even if they do appear, only partial segments may do so. If this factor is considered in developing the algorithm, offsetting operations can work more efficiently. The classification of the edges and their positive offset results are illustrated in Table 3.

Table 3 Positive edge offsets

Classification	Example	Condition	Positive offset result
1. Wire edge		Always	Tubular surface
2. Sharp edge		Always	Half tubular surface
3. Inner edge		For convex wedge	Partial tubular surface
		For concave wedge	Ø
		For smooth wedge (G^1 cont.)	Ø

The positive offset of a wire edge is the sheet model of a full tubular surface, while the offset of a sharp edge or an inner edge is the sheet model of a partial tubular surface or nothing. Note that an inner edge can be adjacent to more than two faces in non-manifold models. In this case, the convexity of the edge cannot be defined. Therefore, in this paper, a new type of object, called 'wedge', is introduced. A wedge represents a corner space bounded by an edge and its two adjacent faces. Actually, a wedge class in the system contains three pointers to an edge and its adjacent two partial faces. Each of the wedges can be classified into convex, concave and smooth. If an edge has a convex wedge, a tubular sheet is generated for that wedge in the edge offset procedure.

A tubular sheet model for the edge offset is generated by sweeping a circular wireframe along the edge curve in ANYSHAPE. In the system, a wireframe model is first generated for the profile of a circular arc, and then it is swept along the edge curve. The procedures to generate a profile wireframe for each case of Table 3 are described in the following subsections.

(1) Profile Wireframes for Wire Edge Offsets

In this case, a full circle is generated for the profile of the sweeping operation. The circle is defined by the following equation:

$$C(t) = E(0) + r\,(\,cos(t)\,\mathbf{n} + sin(t)\,\mathbf{b}\,),\ 0 \le t \le 2\pi \qquad (21)$$

where $E(0)$ is the start position of the edge, $\mathbf{n}$ is the unit normal at $E(0)$, and $\mathbf{b}$ is the binormal calculated by $\mathbf{b} = \mathbf{t} \times \mathbf{n}$ if $\mathbf{t}$ is the unit tangent at $E(0)$.

When creating a wireframe model for this profile, four quadrant edges are constructed for easy manipulation of topological data in ANYSHAPE. The sequence of Euler operators to create this circle is as follows:

MMR → MVS at $C(0)$ → MEV from $C(0)$ to $C(\pi/2)$ → MEV from $C(\pi/2)$ to $C(\pi)$ → MEV from $C(\pi)$ to $C(3\pi/2)$ → MEC from $C(3\pi/2)$ to $C(0)$

(2) Profile Wireframes for Sharp Edge Offsets

As shown in Fig. 11, the wireframe of a semicircle needs to be generated for the profile of the sweeping operation. The semicircle can be defined as follows:

$$C(t) = E(0) + r\,(\,cos(t)\,\mathbf{n}_{PF\_L} + sin(t)\,\mathbf{b}\,),\ 0 \le t \le \pi \qquad (22)$$

where $\mathbf{n}_{PF\_L}$ is the normal of the left partial face in the wedge and $\mathbf{b}$ is the binormal of $\mathbf{t}$ and $\mathbf{n}_{PF\_L}$, which is calculated by $\mathbf{b} = \mathbf{t} \times \mathbf{n}_{PF\_L}$.

Actually, when creating a wireframe model for this profile, two equadrant edges are generated by the following sequence of Euler operators:

MMR → MVS at $C(0)$ → MEV from $C(0)$ to $C(\pi/2)$ → MEV from $C(\pi/2)$ to $C(\pi)$

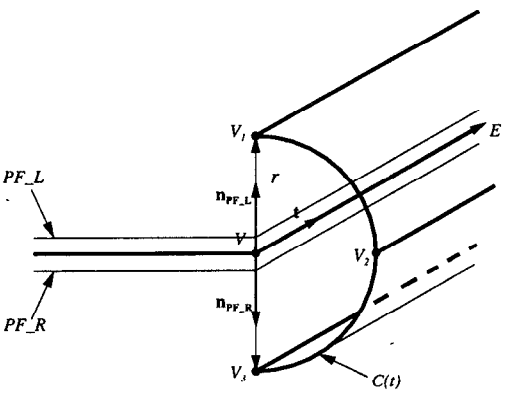


Fig. 11 An edge offset for a sharp edge

(3) Profile Wireframes for Inner Edge Offsets

If there is a convex wedge at an edge and the angle between the left and right partial faces does not change along the edge, a partial tubular sheet model is generated as the edge offset. For the sweeping profile, the wireframe of a circular arc is generated. The arc is defined as follows:

$$C(t) = E(0) + r\,(\,cos(t)\,\mathbf{n}_{PF\_L} + sin(t)\,\mathbf{b}\,),\ 0 \le t \le \theta \qquad (23)$$

where $\mathbf{n}_{PF\_L}$ is the unit normal of the left partial face in the wedge, $\mathbf{b}$ is the binormal of $\mathbf{t}$ and $\mathbf{n}_{PF\_L}$ ($\mathbf{b} = \mathbf{t} \times \mathbf{n}_{PF\_L}$), and θ is the angle from $\mathbf{n}_{PF\_L}$ to $\mathbf{n}_{PF\_R}$.

Since the angle of the arc is less than 180°, a wireframe model is composed of only one edge in ANYSHAPE. The generation procedure is as follows:

MMR → MVS at $C(0)$ → MEV from $C(0)$ to $C(\theta)$

4.4 Generation of Face Offsets

The boundary of the positive face offset is a set of points that are at a distance r from the face. It can also be viewed as a rolling ball surface of radius r when its center moves through all the points on the face. In this paper, normal offsetting is defined as offsetting a face by a given distance r along the face normal. As illustrated in Fig. 12, the rolling ball surface can be divided into three types of segments: spherical surfaces originated from vertices (for example, F_{VI}), tubular surfaces from edges (for example, F_{EI}) and two normal offset surfaces from the face (for example, F_{FI}). Since the spherical and tubular surfaces were already generated as the vertex and edge offsets in Section 4.2 and 4.3, only the normal face offsets need to be generated for the face offsets.

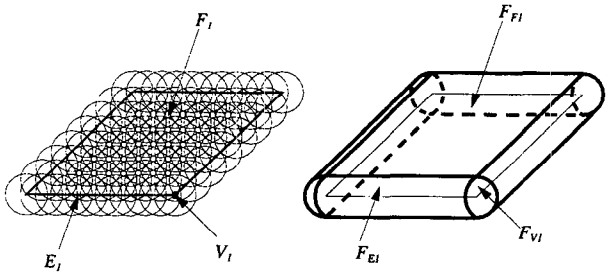


Fig. 12 A positive face offset

According to the number of *void* regions adjacent to a face, the faces are classified into two groups, i.e. , laminar faces and normal faces, and different number of the normal offsets are generated as illustrated in Table 4. Laminar faces are adjacent to two *void* regions, and two offset sheet models are generated in both normal directions of the face. Normal faces are adjacent to one *void* and one *filled* regions, and only one offset sheet model is generated in the normal direction to the void region

Table 4 Positive face offsets

Classification	Example	Positive offset result
1. Laminar face		Two offset faces
2. Normal face		One offset face

If $\mathbf{n}$ denotes the face normal to the void region and F denotes the face, the normal face offset F_o is as follows:

$$F_o = \{ \mathbf{p}_o \mid \exists \mathbf{p} \in F, \mathbf{p}_o = \mathbf{p} + r\,\mathbf{n} \} . \quad (24)$$

If the face surface S is defined by a function of two parameters, u and v, its normal offset surface S_o is as follows:

$$S_o(u,v) = S(u,v) + r\,\mathbf{n}(u,v) , \quad (25)$$

where

$$\mathbf{n}(u,v) = (S_u \times S_v) / \| S_u \times S_v \| . \quad (26)$$

If $C(t)$ denotes the edge curve that constitute the face boundary, its normal offset curve $C_o(t)$ is as follows:

$$C_o(t) = C(t) + r\,\mathbf{n}(t) , \quad (27)$$

where

$$C(t) = S(u(t), v(t)) \quad (28)$$

and

$$\mathbf{n}(t) = \frac{S_u(u(t),v(t)) \times S_v(u(t),v(t))}{\| S_u(u(t),v(t)) \times S_v(u(t),v(t)) \|} . \quad (29)$$

If $\mathbf{p}_V$ denotes the position of a vertex on the face boundary, its normal offset $\mathbf{p}_{Vo}$ is as follows:

$$\mathbf{p}_{Vo} = S_o(u,v) = \mathbf{p}_V + r\,\mathbf{n}(u,v) , \quad (30)$$

where

$$\mathbf{p}_V = S(u,v) . \quad (31)$$

Note that a face offset can be degenerated to a vertex if the face has a spherical surface whose radius is the same as the offset distance. In addition, a face offset can be degenerated to a wire edge if the face has a tubular surface whose radius of curvature is the same as the offset distance. Moreover, a face offset can self-intersect if the face surface does not satisfy the following property:

$$r < min|1/\kappa(u,v)| , \quad (32)$$

where $\kappa(u,v)$ means the curvature of $S(u,v)$. In this case, new edges and vertices should be generated. In this paper, however, the self-intersection of each offset sheet model for edges and faces is not considered in this algorithm as mentioned above and remains as future work. However, this does not mean that this algorithm can not manage the situation that the resulting offset body is self-intersecting. Such a global self-intersection is eliminated automatically by this algorithm, and thus no correction procedure is necessary here.

Face Offset Algorithm

1. Generate a new model using MMR.
2. Check whether or not the face has a spherical surface whose radius is the same as the offset distance. If so, generate a vertex at the center position of the spherical surface using MVS, and then quit this procedure.
3. Check whether or not the face has a cylindrical surface whose radius is the same as the offset distance. If so, generate a wire edge using MVS and MEV, and then quit this procedure.
4. Otherwise, a sheet model for the face offset is generated as follows:
   - *4.1* Collect the edges and vertices that constitute the face boundary. Then, generate the offset curves for the edges with Eq. (27) and the offset points for the vertices with Eq. (30).
   - *4.2* Select a vertex on the peripheral loop L_l of the face. Then, generate a vertex using MVS at the offset position of the selected vertex.
   - *4.3* If the loop L_l has n edges, generate a wire loop of n edges with their offset curves by calling (n-*1*) MEV's and one MEC.

4.4 Generate a face to close the wire loop using MFKC. The offset surface is attached to this face.

4.5 If the original face has hole loops, do the following procedure for each hole loop. Otherwise quit this algorithm.

4.5.1 Generate a bridge edge to connect a vertex on the peripheral loop L_1 with a vertex on the hole loop L_2. Note that the bridge edge should be made not to intersect with any other edges of the hole loop L_2.

4.5.2 Generate edges for the hole loop with the offset curves using MEV's and MEF's.

4.5.3 If any new face created in Step 4.5.2 is an actual hole in the original face, remove it using KFMC to generate a hole.

4.5.4 Remove the bridge edge using KEML.

4.5 Union of the Vertex, Edge and Face Offsets

Once the offset sheet/solid models for the vertices, edges and faces are generated, they are united into a single non-manifold offset model using the non-manifold Boolean operations. Research for Boolean operations has been made by Crocker [1], Gursoz [4], Masuda [12], Kim [7] and many others. Most of them are based on the concept of 'merge and selection' in common. Based on Masuda's algorithm, Kim implemented Boolean operations in ANYSHAPE using Partial Entity Data Structure and the non-manifold Euler operators. As a result, the offsetting operations in this paper use Kim's Boolean operations in order to unite the offsets of the vertices, edges and faces into one. Refer to the reference [7] for its detailed description.

Although this approach works well under the restriction on the surface types, i.e., for planar and quadratic surfaces, it is true that this approach is a brute-force one that will lead to so much computation time and serious numerical errors for the models with freeform surfaces. To overcome this shorcoming, a new approach using the topological relationships is under development.

4.6 Removal of Unnecessary Topological Entities

The united offset model can have topological entities that are within the offset distance r from the original model X. They should be removed to complete the boundary of the non-manifold offset model. If the united offset model is denoted by X_o, the removal algorithm is as follows:

1. Search for the faces, edges and vertices of X_o to be removed. The system calculates the closest distance between each entity of X_o and X. If the distance is less than r, the entity of X_o is marked to be deleted.

2. Delete the marked faces using KFMC or KFR. If the faces are adjacent to the same region in both sides of the face, KFMC is called. Otherwise, KFR is called to delete not only the face but also one of its adjacent regions to merge two regions into one.

3. Delete the marked edges using KEC, KEV or KEMS. KEC is used when the edge is a part of a wire loop. KEV is used when the edge is a strut edge with a free end. KEMS is used when a new isolated vertex is generated by deleting the edge. Note that some marked vertices can be deleted when KEV's are called.

4. Delete the marked vertices, which survived Step 3, using KVS's.

5 VARIATIONS OF THE NON-MANIFOLD OFFSET ALGORITHM

The offset algorithm based on the mathematical definition needs to be modified for more practical offset results for various application areas. In this section, some variational offset algorithms for wireframe and sheet models are discussed.

5.1 Variations for Wireframe Offsetting

If a user wants to obtain solid models for pipelines, he may create wireframe models and then convert them into solids using the offsetting operations proposed in this paper. However, the resulting solids are far from actual product shapes because they have spherical surfaces at the ends of pipelines as shown in Fig. 13(a). Therefore, the offset algorithm needs to be adapted for more practical results as shown in Fig. 13(b). Since the spherical surfaces originated from the offsets of the vertices at the ends of the wireframe, those vertex offsets should be replaced with sheet models for flat discs. On the other hand, if the user wants to obtain sheet models for pipelines, the end-vertex offsets should not be generated.

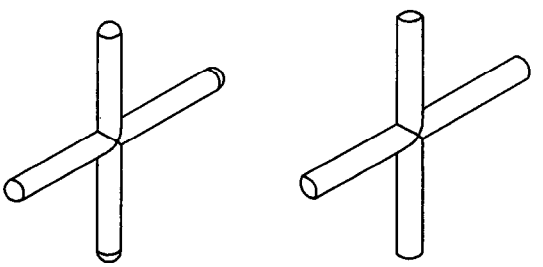


Fig. 13 Wireframe offsets for solid modeling of pipelines (a) Applying the algorithm based on the mathematical definition (b) Applying the adapted algorithm for more practical pipeline modeling

5.2 Variations for Sheet Offsetting

Several researches for converting sheets to solids have been made for easier solid modeling for thin plastic or sheet metal parts. The offsetting operations described above can be used for this purpose. However, since the offset algorithm is based on mathematical definitions, the offset solid of a sheet has the thickness faces with tubular and spherical surfaces as shown in Fig. 14(a). Since this is rarely shown in actual plastic or sheet metal parts, they have to be changed to a model shown in Fig. 14(b). That is, the tubular surfaces have to be replaced with ruled surfaces and the spherical surfaces have to be removed or replaced with planar or freeform surfaces.

The tubular surfaces are generated by sweeping an arc along the sharp edges at the edge offsetting step in Section 4.3.

Therefore, if a line is adopted as the sweeping profile instead of an arc, they are converted to the ruled surfaces immediately. The end points $\mathbf{p}_{v1}$ and $\mathbf{p}_{v2}$ of the line are calculated as follows:

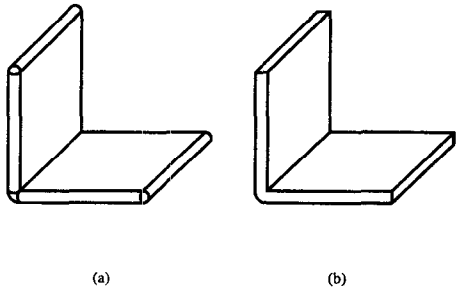
$$\mathbf{p}_{v1} = E(0) + r\,\mathbf{n}_{PF\_L} \quad (33)$$

$$\mathbf{p}_{v2} = E(0) + r\,\mathbf{n}_{PF\_R} \quad (34)$$

The spherical surfaces originate from the offsets of the singular vertices of the sharp edges. One probable algorithm is not to generate vertex offsets for these vertices. Then, some holes may remain on the thickness faces when the offset models for the faces, edges and vertices are united in a single model. In that case, the system detects and closes these holes with new faces. If the boundary of a hole is on a plane, the plane is created and attached to the face. Otherwise, a freeform surface that interpolates the hole boundary is attached to the face. If there are n holes, MFKC's are called $n-1$ times and then one MFR is called finally to close n holes. If the user wants to clean up the resulting offset solid, he can use the tidy operations that merge two neighbor faces or edges with the same geometry and delete isolated vertices and edges.

When a user creates a thin-walled solid model, in many cases, he/she prefers to offset a sheet model for the outer or inner shape of the part in one normal direction rather than to offset in both normal directions. To meet this requirement, one of the end points $\mathbf{p}_{v1}$ and $\mathbf{p}_{v2}$ in Eq. (33) and (34) should be $E(0)$ depending on inner or outer offsetting.

This modified algorithm is more logical and easier than those of Stroud [19], Lee and Kwon [8], or Lim and Lee [11] that are based on solid data structures and modeling functions. However, the algorithm should be designed more precisely considering exceptional cases of thickness faces. This remains as future work.

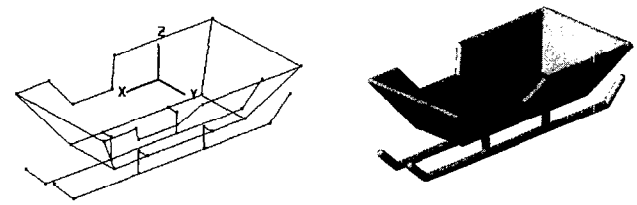


(a) (b)

Fig. 14 Sheet offsets for thin-part modeling (a) Applying the algorithm based on the mathematical definition (b) Applying the modified algorithm for practical thin-part modeling

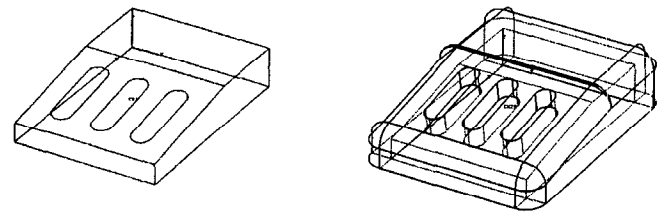
6 CASE STUDY

For case study, as shown in Fig. 15, a non-manifold model for a snow sled is created with wire edges and sheets, and then it is offset using the non-manifold offsetting operation based on the mathematical definition. Fig. 16(a) shows a sheet model for a mouse. The modified offsetting operation for sheet models is applied to the mouse and obtain the offset solid as shown in Fig. 16(b).



(a) (b)

Fig. 15 Offsetting a non-manifold model composed of wireframe and sheet objects (a) A simple non-manifold object (b) A positive offset model



(a) (b)

Fig. 16 Offsetting a sheet model using variational offset algorithm for sheet objects (a) A simple non-manifold sheet object (b) A positive offset model

7 CONCLUSION

This paper introduced offsetting operations in non-manifold geometric modeling, which can be applied to not only wireframes, sheets and solids but also combinations of them in one integrated modeling environment. The mathematical definitions and properties of non-manifold offsetting operations were discussed first, and then the offset algorithm using the non-manifold Euler operators were described in detail. In addition, variations of the offset algorithm for wireframes and sheets were discussed to give more practical offset solids although more effort is needed to formalize these algorithms. The offset algorithms have been implemented in ANYSHAPE, which is a noncommercial non-manifold modeler based on the Partial Entity Data Structure. Since the surface types are restricted to planar and quadratic so far, more work is required to enlarge the surface types. In addition, the non-manifold offset algorithm is applicable to shelling operations that turn a solid into a thin-walled shell of constant thickness because the inner wall is the offset to the outer wall by a given thickness.

REFERENCE

[1] 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.

[2] Farouki, R. T., "Exact Offset Procedures for Simple Solids," Computer Aided Geometric Design, Vol. 2, pp. 257-279, 1985.

[3] 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.

[4] Gursoz, E. L., Choi, Y. and Prinz, F. B., "Boolean Set Operations on Non-manifold Boundary Representation Objects," Vol. 23, No. 1, pp. 33-39, January/February 1991.

[5] Kaul, A., "Minkowski Sums: A Simulation Tool for CAD/CAM", Computers in Engineering, Vol. 1, pp. 447-456, ASME, 1992

[6] Kim, S. H., "Development of Form Feature Based Non-manifold CAD System Supporting Top-down Assembly Modeling," Ph. D. Thesis, Seoul National University, 1994 (in Korean).

[7] Kim, S. H., Lee, K. and Kim, Y. J., "Development of Boolean Operations for CAD System Kernel Supporting Non-manifold Models", Transaction of the SCCE, Vol. 1, No. 1, pp. 20-32, 1996 (in Korean).

[8] Lee, K. and Kwon, B. W., "Efficient Modeling Method of Sheet Objects," Proc. ASME Computers in Engineering Conference, San Francisco, CA, Vol. 1, pp. 437 - 446, August 2-6, 1992.

[9] Lee, S. H. and Lee, K., "Compact Boundary Representation and Generalized Euler Operators for Non-manifold Geometric Modeling," Transaction of the SCCE, Vol. 1, No. 1, pp. 1-19, 1996 (in Korean).

[10] Lee, S. H. and Lee, K., "Sheet Modeling and Transformation of Sheet into Solid Based on Non-manifold Topological Representation," Journal of the Korean Society of Precision Engineering, Vol. 13, No. 7, pp. 100-114 (in Korean).

[11] Lim, H. S. and Lee, K., "Efficient Solid Modeling via Sheet Modeling," Computer Aided Design, Vol. 27, No. 4, pp. 255-262, 1995.

[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] Middleditch, A.E., "The Representation and Manipulation of Convex Polygons", Theoretical Foundations of Computer Graphics and CAD (NATO ASI Series F, Vol. 40), Edited by.Earnshaw, R.A, pp. 211-252, Springer-Verlag, 1988

[14] Pham, B., "Offset Curves and Surfaces: a Brief Survey", Computer-Aided Design, Vol. 24, No. 4, pp. 223 - 229, April 1992.

[15] Rossignac, J. R. and Requicha, A. A. E., "Offsetting Operations in Solid Modelling," Computer Aided Geometric Design, Vol. 3, pp. 129-148, 1986.

[16] Saeed , S. E. O., de Pennington, A. and Dodsworth, J. R., "Offsetting in geometric modeling," Computer-Aided Design, Vol. 20, No. 2, pp. 50-62, 1988

[17] Satoh, T. and Chiyokura, H., "Boolean Operations on Sets Using Surface Data," ACM SIGGRAPH: Symposium on Solid Modeling Foundations and CAD/CAM Applications, Austin, Texas, USA, pp. 119 - 127, June 5-7, 1991.

[18] Spatial Technology Inc, Chapter 9. Shelling, ACIS Optional Husks 5.0, pp. 9-1 ~ 9-7, 1999.

[19] Stroud, I., "Modeling with Degenerate Objects," Computer Aided Design, Vol. 22, No. 6, pp 344-351, 1990.

[20] Yamaguchi, Y. and Kimura, F., "Non-manifold Topology Based on Coupling Entities," IEEE Computer Graphics and Applications, Vol. 15, No. 1, pp. 42-50, January 1995.