Computer-Aided Design 41 (2009) 830–846

Contents lists available at ScienceDirect

Computer-Aided Design

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

Offsetting operations on non-manifold topological models

Sang Hun Lee *

School of Mechanical and Automotive Engineering, Kookmin University, Jeongneung-Dong, Seongbuk-Gu, Seoul, 136-702, Republic of Korea

ARTICLE INFO

Article history:
Received 5 October 2008
Accepted 8 May 2009

Keywords:
CAD
Geometric modeling
Offset
Non-manifold
Wireframe
Sheet
Solid
Sheet thickening
Solid shelling

ABSTRACT

This paper describes non-manifold offsetting operations that add or remove a uniform thickness from a given non-manifold topological model. The mathematical definitions and properties of the non-manifold offsetting operations are investigated first, and then an offset algorithm based on the definitions is proposed and implemented using the non-manifold Euler operators proposed in this paper. In this algorithm, the offset elements of minimal size for the vertices, edges and faces of a given non-manifold model are generated first. Then, they are united into a single body using the non-manifold Boolean operations. In order to reduce computation time and numerical errors, the intersections between the offset elements are calculated considering the origins of the topological entities during union. Finally, all topological entities that are within the offset distance are detected and removed in turn. In addition to the original offset algorithm based on mathematical definitions, some variant offset algorithms, called sheet thickening and solid shelling, are proposed and implemented for more practical and efficient solid modeling of thin-walled plastic or sheet metal parts. In virtue of the proposed non-manifold offset operation and its variations, different offsetting operations for wireframes, sheets and solids can be integrated into one and applied to a wide range of applications with a great potential usefulness.

1. Introduction

1.1. Background and objectives

Offsetting operations add or remove a uniform thickness from a given object. For a positive offset, one adds to an object 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 not only to solid models but also to 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. Offsetting operations cover a wide range of potential applications. 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 [1]. Solid offsetting has been used for tolerance analysis, clearance testing, NC tool path generation [2], collision-free path planning for robots [3], rapid prototyping [4], constant-radius rounding and filleting and so on [5].

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 were developed separately and used in their own restricted application areas. In recent years, however, non-manifold geometric modeling systems have been developed and more widely spread in industries, which can manipulate all wireframes, surfaces, solids, and their mixtures with a single unified topological representation. Therefore, if we develop offsetting operations for non-manifold topological (NMT) models, three types of operations can be integrated into one in a single environment to be used in many application areas. For instance, 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]. In addition, if we give some variances to the original offset algorithm, thin-walled solid models for plastic or sheet metal parts can be created easily from the sheet models.

To take great advantages of non-manifold offsetting, in this paper, the mathematical definitions and properties of the non-manifold offsetting operations are studied first, and then the positive and negative offset algorithms are proposed and implemented using the non-manifold Euler and Boolean operations. In addition, some variant offset algorithms called sheet thickening and solid shelling are proposed and implemented for more practical and efficient solid modeling of thin-walled plastic or sheet metal parts. Note that this work is mainly focused on topological operations for offset models, although most of the previous research has

* Tel.: +82 2 910 4835; fax: +82 2 910 4839, 82 2 916 0701.
E-mail address: shlee@kookmin.ac.kr.

0010-4485/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2009.05.001

been concentrated on geometric calculations for developing offset curves and surfaces.

The remainder of this paper is organized as follows. Section 1.2 surveys the related work. Section 2 describes the formal definitions and mathematical properties of non-manifold offsetting. Section 3 presents a set of Euler operators for non-manifold models based on extended Euler Poincare formula. Section 4 describes an offset algorithm for non-manifold models, which is implemented using a set of non-manifold Euler and Boolean operations. Section 5 presents variant offset algorithms for wireframe and sheet models, which include sheet thickening and solid shelling functions to allow users to create more practical offset solids from sheets and solids. Finally, Section 6 presents the conclusions of this paper.

1.2. Related work

A lot of research related with offsetting has been carried out for over three hundred years, and they 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. Pham [7] carried out a literature survey on offset curves and surfaces up to 1992, and Maekawa [8] overviewed the literature after 1992 focusing on five active areas of research on offsets: representing exact offsets in Bézier/B-spline format, approximations, self-intersections, geodesic offsets and general offsets. 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 concentrates on the description of topological operations for offsetting a non-manifold model. Although no publications on non-manifold offsetting was found, 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 [5] 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 [9] 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, the topology of the solid is first resolved as faces, edges and vertices. For each face, a face offset element is generated by offsetting the face by a vector $r\,\mathbf{n}$, where $\mathbf{n}$ is the outward normal to that face, and r is the offset distance. For each edge, an 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 complete, 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. [10] 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. Their definition of an offset is equivalent to that given by Rossignac and Requicha [5]. 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 [11] 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, he 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, these sets 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 [12] proposed algorithms for offsetting and shelling operations on B-rep solids, and also implemented them with the modeling capabilities of SolidDesigner. In his offset algorithm, an offset solid is generated as follows. 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 the substitution of the geometric entities of a given solid with the offset ones and then the blending of the concave or convex edges. However, the offset solid can show, in geometric substitution, topologically irregularity, for which he did not suggest any topology correction procedures.

Recently, Kumar et al. [13] proposed an approach for the automatic offset of a NURBS B-Rep, which can be used for a class of manifold B-Reps. The approach offsets each of the trimmed surfaces (faces) of the B-Rep and then removes the gaps and intersections between offset faces automatically, if any. The offset B-Rep is then created by sewing all the updated offset faces. The approach can generate both positive and negative constant offsets of a B-Rep. The approach works under the assumption that the number of faces in both base and offset B-Reps is the same. It further assumes that the faces are at least G^1 continuous and non-self-intersecting after constant offset. The approach has been applied to composite laminate modeling to make offsets of many lay-up surfaces.

The commercial geometric modeling kernels such as Parasolid and ACIS also provide functionality for solid offsetting. ACIS [14] provides the body offset operation that can be applied not only to the whole body but also to the specific faces of the body. In both cases, the surfaces of the selected faces are replaced with offset surfaces by using the tweak operation. The tweak operation replaces the surface under a face with any other surface supplied by the user, provided that the surface intersects appropriately and that necessary topology changes are supported. However, faces with radial surfaces cannot be offset. If they exist, they are removed and the resulting wound is healed by the surrounding face surfaces. Parasolid [15] also provides the offsetting operations analogous to

ACIS's. If the offset surfaces do not meet, the underlying surfaces are extended to enable the offsets to intersect. Recently, however, a new functionality for the automatic detection and repair of self-intersecting geometry has been included. If the creation of offset surfaces introduces self-intersections, the system removes such self-intersections, and heals any resulting holes either by extending and intersecting neighboring offset surfaces, or by filling them with an approximate surface that joins the neighboring regions within the given offset tolerance.

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. In this method, a sheet model is first created for one side of the part or for a medial surface of the part, and then a thin-walled solid is generated by adding volume to the sheet by a given thickness. This method is usually called sheet thickening, which is very useful in modeling plastic or sheet metal parts. 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 wall. In sheet thickening approaches, the sharp edges in a sheet body are converted to the thickness faces.

Stroud [16] 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 verification and correction methods for self-intersections of the converted solid.

Lee and Kwon [17] proposed another sheet modeling and transformation approach, which was revised by Lim and Lee [1]. They adopted 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 with respect to the storage of 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 find a way to overcome these drawbacks, they investigated failure cases and suggested topological correction methods for unacceptable solids, although such cases are limited, and they did not consider the self-intersection problem. 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 the 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 [16] and Lee and Kwon [17] 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, even though sheets are fundamentally non-manifold objects. This deficiency makes it difficult to develop algorithms for sheet modeling and thickening capabilities.

The current solid modeling systems usually provide sheet thickening operations which can convert a sheet model to a solid model by offsetting it in one or both normal directions [14]. For example, ACIS provides the *api_sheet_thicken* function, and Parasolid offers the *PK_BODY_thicken_3* function. However, these functions have several limitations currently. First, the sheet must be manifold, single lump, and single shell. Next, the normals of adjacent faces must be consistent. Therefore, it cannot generate an offset for an arbitrary non-manifold body. For example, this T-shape body cannot be offset exactly. In addition, ACIS cannot fix the self-intersecting surfaces either, while Parasolid can do recently. Similar limitations exist in the existing commercial systems.

There is another way to model a thin-walled solid efficiently through a method called solid shelling. The solid shelling operation creates a thin-walled solid model by digging out the inside volume from the solid model of the outside shape. Although this operation is not suitable for modeling sheet metal parts with many bending and hole features, it is very useful for modeling plastic parts in the shape of bowls.

Lee and Lee [18] 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 [12] 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 [18] except that it checks to correct the self-intersection of the shelled body and rounds the convex edges and vertices of the offset solid.

Recently, in rapid prototyping, the shelling method has received much attention because it can reduce the building time and expensive RP material consumption significantly by constructing a hollowed prototype instead of a solid model. For hollowing a solid model, several methods have been developed, and they can be classified into four categories [19]: spatial enumeration techniques such as the octree and the voxel model [20–22]; constructive solid geometry (CSG) offsetting methods [4]; curve offsetting methods [19,23]; and surface offsetting methods [24,25]. In the enumeration methods, the inner wall of a thin-shell solid model is obtained by using a sub-boundary octree [21] or voxel elements [20,22]. These methods are computationally effective, but can cause an internal staircase effect. In the CSG offsetting method [4], a thin-shell solid is obtained by subtracting the original solid from its offset counterpart. However, this method is only applicable to CSG solids. In the curve offsetting method proposed by Ganesan and Fadel [23], by slicing the original part, external cross-sectional curves are obtained first, and then two-dimensional curve offsetting is carried out to find the internal cross-sectional curves. Finally, the interior surface is constructed from the internal curves. Since the curve offsetting method employs two-dimensional curve offsetting instead of three-dimensional surface offsetting, this approach is relatively easy to implement. However, it cannot guarantee uniform wall thickness because the wall thickness depends on the surface normals. To overcome this drawback, Park [19] developed a new method to generate internal contours directly from the external contours. The sum of the circle swept volumes of

external contours represents the offset model. It is possible to compute an internal contour of a layer by slicing the circle swept volumes affecting the layer. Actually, it is not necessary to compute the actual circle swept volumes because we can generate the sliced curves with a simple combination of two-dimensional geometric operations. Since the algorithm is based on well-known two-dimensional geometric algorithms, it is efficient and easy to implement. As for the surface offsetting methods, Koc and Lee [24] proposed a non-uniform vertex offsetting method based on an averaged surface normal method to hollow out a solid model, and Qu and Stucker [25] proposed a vertex offset method, in which topology information is built first and then the vertex offset is calculated using the weighted sum of the normals of the adjoining facets. However, in the solid hollowing method for RP processing, the input geometric model is a polyhedral solid model described in the STL format, and the output is also a polyhedral model offset by a given thickness. Therefore, this method is not adequate for obtaining the exact offset of a general solid model with quadratic and freeform surfaces.

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, which 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 a Euclidean cell complex [26].

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

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

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

where e_λ denotes an n-cell, Λ a universal set of n-cell indices, $dim(e_\lambda)$ 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

2.2.1. Definition of non-manifold offsets

If X denotes a non-manifold model defined as a three-dimensional 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:

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

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.

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).

2.2.2. Mathematical properties of non-manifold offsets

Various mathematical properties of solid offsets were discussed by Rossignac and Requicha [3]. 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 three-dimensional Euclidean cell complex are algebraically closed under offsetting operations. That is, if X is a three-dimensional Euclidean cell complex, then its offsets $X \oplus r$ and $X \ominus r$ are also three-dimensional 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, the positive and negative offsetting operations are not generally commutative, and the two operations should not be thought of as inverses because they have the following properties:

$$(X \ominus r) \oplus r \subset X \subset (X \oplus r) \ominus r. \tag{7}$$

Actually, we can obtain the rounding and filleting effects of non-manifold models by combining the 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 [3].

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 $d(\mathbf{p}_o, X)$ denotes the distance from a point outside X, $\mathbf{p}_o$, to the closest point on X, this relationship can be written as follows:

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

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

where

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

and

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

The boundary of the offset model has the following property, as in a solid offset [3]:

$$\partial(X \oplus r) \subset \partial(\partial X \oplus r). \tag{12}$$

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 a given model X. Since the boundary of the resulting offset model $\partial(X \oplus r)$ is included in $\partial(\partial X \oplus r)$, $\partial(X \oplus r)$ 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.

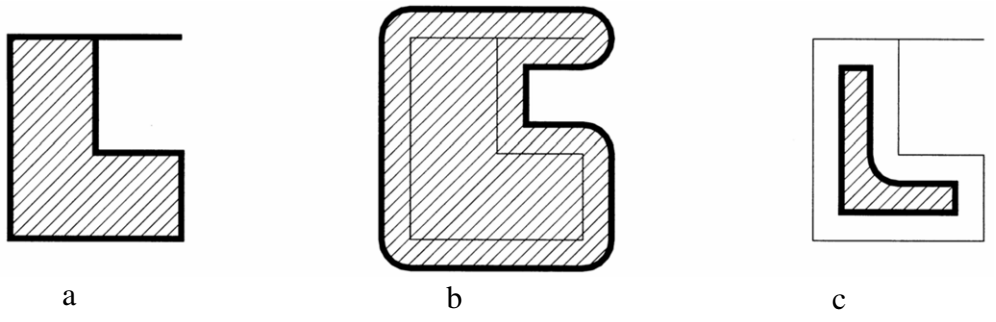

Fig. 1. Non-manifold offsetting operations: (a) a simple non-manifold object composed of a sheet and an L-shaped solid; (b) a positive offset; (c) a negative offset.

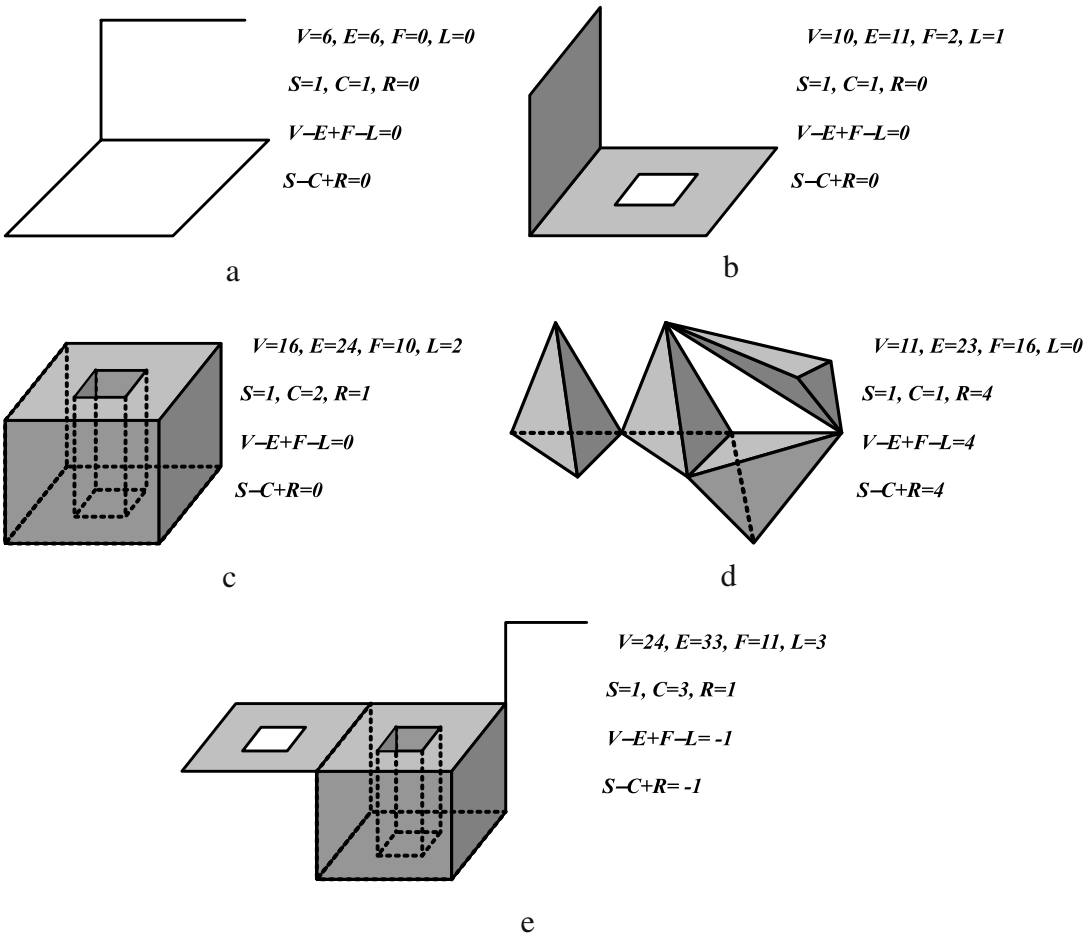

Fig. 2. Examples of the application of the extended Euler–Poincare formula.

3. Non-manifold topological operators

The algorithms for the offsetting operations proposed in this paper are described using a set of the basic non-manifold topological (NMT) operators including the extended Euler operators for non-manifold objects. Euler operators in B-rep solid modeling guarantee the integrity of the model, insulate geometric modeling functionality of higher level from the specifics and complexities of underlying data structures, and allow easy implementation of undo commands, as each operator has its inverse operator [27]. To take advantage of these useful features, considerable effort has also been made to derive an Euler–Poincare formula generalized for three-dimensional non-manifold models, and define a set of Euler operators based on the formula [26,28–32].

This paper proposes a set of NMT operators that includes a set of Euler operators based on the following equation, which is an extended Euler–Poincare formula for three-dimensional non-manifold models proposed by Yamaguchi and Kimura [28].

$$V - E + F - L = S - C + R \tag{13}$$

where V, E, F, L, S, C, and R are the numbers of vertices, edges, faces, hole loops, void shells, non-manifold cycles, and regions, respectively. The non-manifold cycle can be interpreted as an independent cycle that is not converted to a face in a wireframe composed of vertices and edges, or a circle that cannot be retracted to a point on a surface. Note that the infinite region is not counted in this formula. Fig. 2 shows some examples to which Eq. (13) is applied.

Table 1
Non-manifold Euler operators and their base vectors.

Category	Name	Meaning	V	E	F	L	S	C	R
Basic NMT operators	MMR	Make model and region							
	KMR	Kill model and region							
Basic Euler operators	MVS	Make vertex and shell	1	0	0	0	1	0	0
	KVS	Kill vertex and shell	−1	0	0	0	−1	0	0
	MEV	Make edge and vertex	1	1	0	0	0	0	0
	KEV	Kill edge and vertex	−1	−1	0	0	0	0	0
	MEC	Make edge and cycle	0	1	0	0	0	1	0
	KEC	Kill edge and cycle	0	−1	0	0	0	−1	0
	MFKC	Make face, kill cycle	0	0	1	0	0	−1	0
	KFMC	Kill face, make cycle	0	0	−1	0	0	1	0
	MFR	Make face and region	0	0	1	0	0	0	1
	KFR	Kill face and region	0	0	−1	0	0	0	−1
	MVL	Make vertex and loop	1	0	0	1	0	0	0
	KVL	Kill vertex and loop	−1	0	0	−1	0	0	0
Auxiliary Euler operators	SEMV	Split edge, make vertex	1	1	0	0	0	0	0
	JEKV	Join edge, kill vertex	−1	−1	0	0	0	0	0
	MEF	Make edge and face	0	1	1	0	0	0	0
	KEF	Kill edge and face	0	−1	−1	0	0	0	0
	KEML	Kill edge, make loop	0	−1	0	1	0	0	0
	MEKL	Make edge, kill loop	0	1	0	−1	0	0	0
	KEMS	Kill edge, make shell	0	−1	0	0	1	0	0
	MEKS	Make edge, kill shell	0	1	0	0	−1	0	0

Some terms needed for the description of Euler operators are defined as follows. A manifold edge is an edge with two incident faces. A non-manifold edge is an edge that is not a manifold edge. A wire edge is a non-manifold edge that is not associated with any face. A sharp edge or a lamina edge is a non-manifold edge that is associated with only one face. If one of the vertices of an edge is adjacent to no other edge, the edge is called a strut edge. A strut edge can be either a manifold edge or a non-manifold edge. An isthmus edge is a manifold edge that is adjacent to the same face and its vertices are adjacent to other edges. Usually an isthmus edge connects an inner loop to another loop of a face. A single-vertex shell is a shell that consists of only an isolated vertex. A single-vertex loop is a loop that consists of only an isolated vertex.

All of the NMT operators in our system are listed in Table 1. According to the naming convention for Euler operators, the following abbreviations are used to represent each operation and topological entity: M (make), K (kill), S (split), J (join), V (vertex), E (edge), F (face), L (hole loop), S (void shell), C (non-manifold cycle), and R (region). As shown in Table 1, they are categorized into three groups. The first group is a pair of the basic NMT operators for initially creating a model and its infinite region, and finally deleting them. The second group is six pairs of basic Euler operators, which are a minimal set of independent Euler operators derived from the formula. Since there are six independent variables in Eq. (13), at least six independent Euler operators and their six inverse operators are required to manipulate the topological structure of non-manifold models. The third group is four pairs of Euler operators supplementing the six pairs of basic operators and facilitating the implementation of high-level modeling operations. The details of these operators including the specification of the input and output arguments are described with figures in the following subsections.

Our kernel modeler was implemented based on the Partial Entity Structure [33], which is a compact but efficient non-manifold boundary representation proposed by the authors. Since all of the high-level modeling operations, such as sweeping and Boolean operations, have been implemented using this set of Euler operators in order to build our kernel modeler, this set has been verified to be sufficient for the development of a non-manifold modeler. The non-manifold offsetting operations suggested in this paper were also implemented using the Euler operators listed in Table 1.

MMR (Model ***M*, Region ** *R*)
KMR (Model **M*)

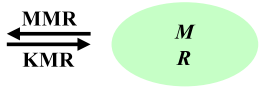

Fig. 3. MMR and KMR operators.

MVS (Region **R*, Vertex ***V*, Shell ****S*, Point **Pt*)
KVS (Shell **S*)

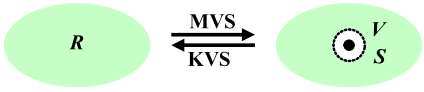

Fig. 4. MVS and KVS operators.

3.1. Basic NMT operators

MMR, KMR

The operator MMR creates, from scratch, an instance of the model, *M*, that has just one infinite region, *R*, as illustrated in Fig. 3. It is the first operator used in any topology construction. The inverse of MMR is called KMR, and it destroys an instance of the model identical to that created by MMR. The input model for KMR should include only one infinite region without any void shell. The C++ style of the function prototypes for MMR and KMR are shown below. A single pointer argument denotes an input whereas a double pointer argument denotes an output.

3.2. Basic Euler operators

MVS, KVS

The operator MVS creates a single-vertex shell, *S*, at a given position, *Pt*, in a given region, *R*, as described in Fig. 4. The isolated vertex created by MVS, *V*, can be used as a starting point for subsequent construction of additional topological features. The inverse of MVS is KVS, which destroys the specific single-vertex shell. The effect of MVS and KVS is depicted in Fig. 4.

MEV, KEV

The operator MEV creates a new vertex, V_2, first, and then creates a new edge, *E*, connecting V_2 with a given vertex, V_1, as

Fig. 5. MEV and KEV operators.

Fig. 6. MEC and KEC operators.

Fig. 7. MFKC and KFMC operators.

illustrated in Fig. 5. The geometry of the edge E is given as an input argument Cv. If the new edge is a wire edge as shown in Fig. 5(a), the shell S containing the edge E is set as the input argument *Parent*. On the contrary, if the edge E is a strut edge as shown in Fig. 5(a), the loop L including the edge E is set as *Parent*. The inverse of MEV is KEV, which destroys the specific edge and vertex. Here, the edge to be deleted should be a strut edge.

MEC, KEC

The operator MEC creates a new wire edge, E, joining two given vertices, vertex V_1 and V_2, to form a non-manifold cycle, as illustrated in Fig. 6. The newly generated edge, E, becomes a part of the specific shell, S. The geometry of the new edge is given as an input argument Cv. The inverse operator of MEC is KEC, which destroys the specific wire edge, E, together with a non-manifold cycle. The edge to be deleted should not be the only pass linking its two end vertices. For example, there are two passes between V_1 and V_2: the one is the edge E, and the other is two remaining edges. If there is only one pass and it is the edge to be deleted, then a new shell is created. The effect of MEC and KEC is depicted in Fig. 6.

MFKC, KFMC

The operator MFKC creates a new face, F, bounded by the specific edges in the list, *E_list*, as illustrated in Fig. 7. As a result, the non-manifold cycle composed of the edges disappears. The input edges as well as the output face belong to the specific shell, S. The geometry of the new face is given as an input parameter *Sf*. The inverse operator of MFKC is KFMC, which destroys the specific face, F, and creates a new non-manifold cycle composed of the boundary edges of F. The effect of MEC and KEC is described in Fig. 7. (OK)

MFR, KFR

The operator MFR creates a new face, F, and a new region, R, from the specific edges in the *E_list*. The region of the specific shell S is separated into two by adding a new face. Some void shells of the original region may be transferred to the new region according to the in/out test. The inverse operator of MFR is KFR, which merges two adjacent regions by destroying the face between them. The void shells of the deleted region, R, are attached to the other surviving region. The effect of MFR and KFR is depicted in Fig. 8.

Fig. 8. MFR and KFR operators.

MVL, KVL

The operator MVL creates a new vertex and a new loop at the same time. As shown in Fig. 9, the loop L is composed of an isolated vertex V, and belongs to the face F. The location of the new vertex is given with an input argument *Pt*. The inverse of MVL is KVL, which destroys the specific single-vertex loop.

Fig. 9. MVL and KVL operators.

3.3. Auxiliary Euler operators

Fig. 10. SEMV and JEKV operators.

SEMV, JEKV

The operator SEMV creates a new vertex, V, and a new edge, E_2, by splitting a given edge, E_1, at a given position, *Pt*, as shown in Fig. 10. Its inverse operator is JEKV, which merges two edges into one. The specific vertex and edge are deleted after JEKV is executed. The edges may be either manifold or non-manifold edges.

MEF, KEF

The operator MEF creates a new edge, E, and a new face, F, by subdividing a given loop, L, by joining two vertices, V_1 and V_2, as illustrated in Fig. 11. The operator attaches the inner loops of the face to either a new face or the other face according to the in/out test result. The inverse operator of MEF is KEF, which merges two adjacent faces by destroying the edge between them, E, and the specific face, F. The inner loops of the deleted face are automatically attached to the other surviving face.

KEML, MEKL

The operator KEML splits a given loop, L, into two new ones, L_1 and L_2, by removing an isthmus or strut edge, E, as illustrated in Fig. 12. This operator is exactly the same as the corresponding Euler operator for solid models. If the edge E is a strut edge, a single-vertex loop appears as a result. The inverse operator of KEML is MEKL, which merges two loops, L_1 and L_2, by joining two vertices, V_1 and V_2, which belong to two different loops of the same face.