PERGAMON

Computers & Graphics 27 (2003) 943–961

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Technical section

Heterogeneous object modeling through direct face neighborhood alteration

Xiaoping Qian[a,*], Deba Dutta[b]

[a] Inspection and Manufacturing Technologies, GE Global Research, KWD 250, One Research Circle, Niskayuna, NY 12309, USA
[b] Department of Mechanical Engineering, The University of Michigan, Ann Arbor, MI 48109-2125, USA

Abstract

Two recent advances—the use of functionally gradient materials in parts and layered manufacturing technology—have brought to the forefront the need for design and fabrication methodologies for heterogeneous objects. However, current solid modeling systems, a core component of computer-aided design and fabrication tools, are typically purely geometry based, and only after the modeling of product geometry, can a part's non-geometric attributes such as material composition be modeled. This sequential order of modeling leads to unnecessary operations and over-segmented 3D regions during heterogeneous object modeling processes.

To enable an efficient design of heterogeneous objects, we propose a novel method, *direct face neighborhood operation*. This approach combines the geometry and material decisions into a common computational framework as opposed to separate and sequential operations in existing modeling systems. We present theories and algorithms for direction face neighborhood alteration, which enables direct alteration of face neighborhood before 3D regions are formed. This alteration is based on set membership classification (SMC) and region material semantics. The SMC is computationally enhanced by the usage of topological characteristics of heterogeneous objects. After the SMC, boundary evaluation is performed according to the altered face neighborhood. In comparison with other solid modeling methods, the direct face neighborhood alteration method is computationally effective, allows direct B-Rep operations, and is efficient for persistent region naming. A prototype system has been implemented to validate the method and some examples are presented.

Keywords: Neighborhood; Heterogeneous object modeling; Set membership classification

1. Introduction

Heterogeneous objects are objects composed of different constituent materials. In these objects, multiple material properties from different constituent materials can be synthesized into one part. Consequently, these objects offer new material properties and multiple functionalities that cannot be obtained otherwise. Two recent advances—use of functionally gradient materials in parts and layered manufacturing technology—have brought to the forefront the need for design and fabrication methodologies for heterogeneous objects [1–5].

The current solid modeling systems, as a core component of CAD/CAM/CAE system, have typically been purely geometry based. Consequently, the modeling of non-geometric product attributes such as material composition, and the modeling of geometric structures have been separated. After a residing region's geometric and topological structures have been formed, the material modeling is conducted. Then the entire object's geometric model is "regularized" according to the material modeling result. This sequential order of

*Corresponding author. Tel.: +1-518-387-4610; fax: +1-518-387-5752.

E-mail addresses: qian@research.ge.com (X. Qian), dutta@engin.umich.edu (D. Dutta).

0097-8493/$ - see front matter
doi:10.1016/j.cag.2003.08.001

geometric operations and material operations leads to unnecessary operations and over-segmented 3D cells for heterogeneous object modeling.

Even though research on modeling and representation schemes for heterogeneous objects has been under way, the work has been primarily focusing on representation of heterogeneous objects, not on design methods. To enable an efficient design of heterogeneous objects, a constructive design method is desired [4].

Given heterogeneous objects $A = \{A_1|^* A_2|^* \dots |^* A_m\}$ and $B = \{B_1|^* B_2|^* \dots |^* B_n\}$ and the constructive operator, the resultant solid needs to be formed. It essentially involves two tasks:

(1) to determine the geometric boundary of A and B that appears in the resultant solid C (*Geometric Boundary Evaluation*) and
(2) to organize the resultant faces into regions and to associate material function m_i to each region g_i (*Material Region Forming*).

In this paper, we propose a novel method, *direct face neighborhood alteration*, to fulfill these two tasks. With two-sided face neighborhood operations, a designer can do concurrent geometric and material operations as opposed to sequential operations in existing methods. This method enables face neighborhood change before 3D regions are formed. It directly alters the face's two-sided neighborhood according to set membership classification (SMC) and material semantics. That is, during the constructive operations, this method concurrently conducts geometric and material operations.

In the remaining of this paper, Section 2 reviews the previous research pertaining to solid modeling. Section 3 then presents the constructive operations for heterogeneous objects. Section 4 presents the direct face neighborhood alteration method for constructive operations. In Section 5, we present an enhanced SMC algorithm utilizing the special topological characteristics of heterogeneous objects. Section 6 gives some examples from the implementation of the direct face neighborhood alteration method. Section 7 compares the face neighborhood alteration with current cellular object modeling method, and describes some possible extensions to face neighborhood alteration method. Section 8 concludes this paper.

2. Literature review

2.1. Representation schemes

Many representation schemes have been developed to represent solids. To represent a solid model, manifold solids and *R*-sets were first proposed to represent solid objects [6,7]. A radial-edge data structure is another data structure for modeling non-manifold solids [8]. For conventional feature modeling, the use of a non-manifold structure was initially proposed in [9]. Selected geometric complex (SGC) is a non-regularized non-homogeneous point set represented through enumeration as union of mutually disjoint connected open cells [10]. Constructive non-regularized geometry (CNRG) was also proposed to support dimensionally non-homogeneous, non-closed point sets with internal structures [11]. A graphic object algebra-based boundary representation was proposed for polygonal heterogeneous solids [12]. A generalized maps boundary representation was proposed in [13]. Middleditch et al. presented mathematics and formal specification for mixed dimensional cellular geometric modeling [14]. Cellular model provides a geometric basis for heterogeneous object modeling. However, the current practice of cellular object modeling has been inept for heterogeneous object modeling. The implementation of cellular object modeling tends to separate the geometric operation from the volume (3D cell) attribute operation. That is, a maximum number of 3D cells are generated first, followed by the propagation of volume attributes, i.e., material composition in the context of heterogeneous objects. Section 7 gives a step-by-step comparison of modeling heterogeneous objects using a conventional cellular modeling method and the proposed face neighborhood alteration method.

Recently, several new representation schemes have been proposed for representing heterogeneous objects. Kumar and Dutta proposed that *R-m* set be used to represent heterogeneous objects [15]. Jackson et al. proposed another modeling approach based on subdividing the solid model into sub-regions and associating analytic composition blending functions with each region [10]. Qian and Dutta proposed feature methodologies for heterogeneous object realization [3–5]. Other modeling and representation schemes using voxel model, distance functions or texturing have also been proposed [16–18].

2.2. Boolean operations for homogeneous solid

Boolean operations for homogeneous solids typically include the following stages: intersection, set membership classification, and boundary update (discarding unnecessary portions and re-organizing of the *B*-rep structure) [6,19,20]. The intersection stage involves the intersecting of the boundaries of two solids. The resultant intersection edges divide each solid boundary into different portions. Each of the portions is then classified against the other solid. The set membership classification refers to the classification of one set against the other set. The classification result can be divided into three categories: *in*, *out* and *on* [21].

Table 1
Boundary classification for homogeneous objects

Set operation	Boundary classification
Union	AoutB|BoutA
Intersection	AinB|BinA
Difference	AoutB|$(\text{BinA})^{-1}$

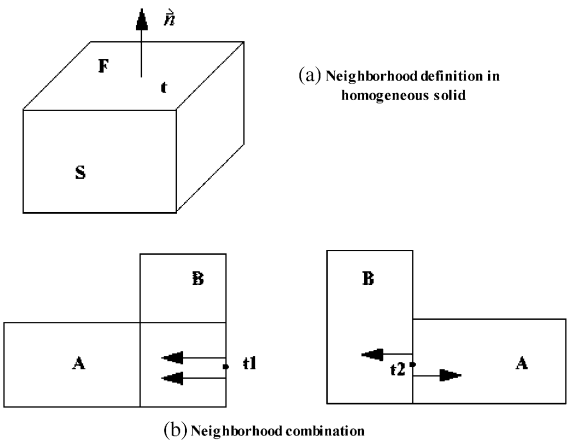


Fig. 1. Neighborhood combination eliminates on/on ambiguity for homogeneous solid. (a) Neighborhood definition in homogeneous solid. (b) Neighborhood combination.

According to the Boolean operation type, some portions are discarded and final part boundary is a collection of the remaining portions. Table 1 lists the collection for different Boolean operations. Note, $(\text{BinA})^{-1}$ means the normal direction of the face is reversed.

Special processing is necessary for situations where points that lie on boundaries of both primitives [19,21,22]. In a typical modeling system, the neighborhood concept is used to facilitate boundary classification for the on/on cases. It is represented by a surface normal, and a side bit, which indicates that the solid's material is either locally on the side toward the normal direction or on the opposite side (Fig. 1a). For example, in Fig. 1b left, point $t1$ has two neighborhoods in the same direction—one from solid A and one from solid B—so the union of the neighborhood is still a one-sided face neighborhood. The difference of neighborhood is *empty*. However, in Fig. 1b right, the point $t2$ has two neighborhoods having opposite directions. So the union of the two neighborhoods leads to a *full*, while the difference is still the one-sided face neighborhood. Therefore, this neighborhood combination eliminates the on/on ambiguity for set membership classification in homogeneous solid modeling.

3. Constructive operations for heterogeneous objects

In this paper, we adopt an *R-m* set as the working representation scheme for heterogeneous objects since it is the most conversed one to us. That is, an *R-m* set (g, m) is used as the building block for the constructive design. For an *R-m* set $A(g, m)$, $m(A)$ gives the material information m, $g(A)$ gives the *R-m* set geometry. " $|^*$" is the *regularized gluing* operation [11].

To support a constructive design of heterogeneous objects, we extend the radial-edge graph (Fig. 2) to represent the geometry of heterogeneous objects. Radial-edge graph data structure is widely used in commercial solid modeling packages and is also the representation scheme used in STEP ISO10303 [23]. In this extended data structure, each region has its material composition representation and each face use has neighborhood information, which contains a pointer pointing to material representation.

3.1. Constructive operations for heterogeneous object design

Constructive operations form the basis of feature-based design. In compliance with form feature classification in STEP, we propose two corresponding constructive operations: additive and subtractive [4]. The operation type reflects the point set change of an object. In addition, we add the partition operation for the convenience of substituting a sub-region's material composition. Each material volume can be thought of as a form feature volume plus the material composition in the region [4]. A compound feature (building block), consisting of more than one *R-m* set can also be defined, i.e., a finite collection of *R-m* sets, (g_1, m_1), $(g_2, m_2), \ldots, (g_n, m_n)$, each consisting of a material

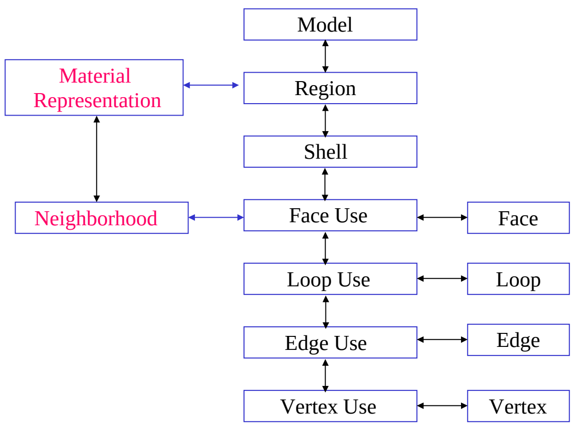


Fig. 2. Extension to the radial edge structure with material representation.

volume. The three constructive operations are defined mathematically as

1. Additive operation

$$(g_1, m_1) + (g_2, m_2) = (g_1 - g_2, m_1)|^*(g_2 - g_1, m_2)|^*(g_1 \cap g_2, m_1 \otimes m_2).$$

2. Subtractive operation

$$(g_1, m_1) - (g_2, m_2) = (g_1 - g_2, m_1).$$

3. Partition operation

$$(g_1, m_1)/(g_2, m_2) = (g_1 - g_2, m_1)|^*(g_1 \cap g_2, m_1 \otimes m_2).$$

To eliminate possible material composition ambiguity in intersecting regions, we introduce material priority tag p to each material volume. That is,

$$m_1 \otimes m_2 = \begin{cases} m_1 & \text{if } p_1 > p_2, \\ m_2 & \text{if } p_1 < p_2, \\ m_1 \oplus m_2 & \text{if } p = p_2. \end{cases}$$

Note, here $m_1 \oplus m_2$ is a user-defined interpolation function. It could be $a_1 \cdot m_1 + (1 - a) \cdot m_2$, $a \in (0, 1)$, or any other form. The $m_1 \oplus m_2$ function has been particularly useful for applications like doping, and implanting, where material volume is "contaminated" by some exotic material.

The partition operation functions the same as additive operations over the intersection region (g_1, g_2), but it is not applicable to the region outside of g_1. This partition operation is used extensively for heterogeneous object modeling when material functions are imposed on a given geometry domain.

We refer to the material and the priority tag of a region as the region's material semantics. It indicates material survival rules over the intersecting regions. Fig. 3 lists the three types of operations and their semantics. Clearly, the part $C = A \otimes B$ depends on the feature type (operation), and each region's materials and the priority tag. Note, we use the symbol "$\otimes$" to represent the three constructive operation types: additive, subtractive and partition, when the operation attending arguments, A and B, are both a collection of R-m sets (g, m). When both arguments are material compositions as in $m_1 \otimes m_2$, the symbol "$\otimes$" represents material semantics in accordance with the material priority tag p.

These constructive operations can be easily customized for many specific applications, e.g., design by composition for layered manufacturing [24], steel bar partially inserted in concrete bar, and MEMS process simulation.

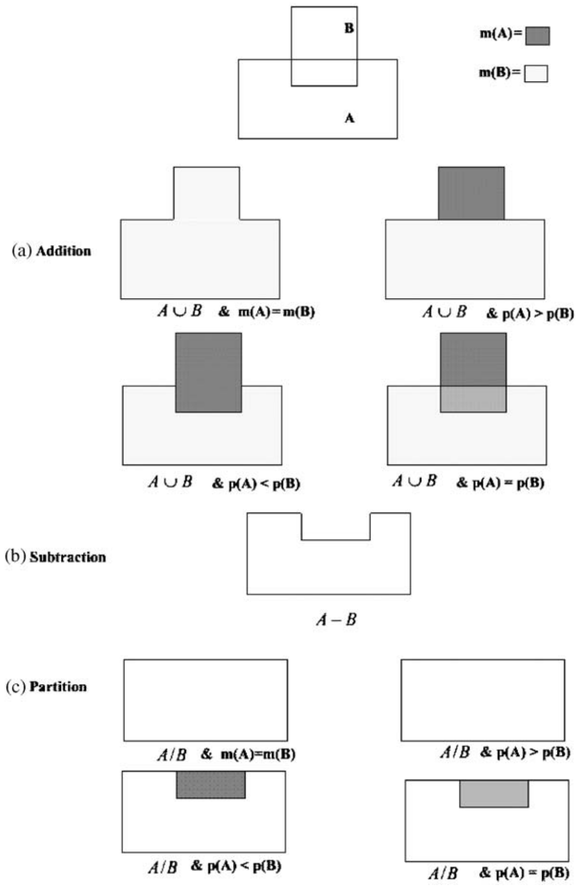


Fig. 3. Generic constructive operations for heterogeneous objects: (a) addition; (b) subtraction; and (c) partition.

3.2. Problem analysis

For the constructive or feature-based design, the modeling tasks in heterogeneous object modeling are different from the homogeneous set operations.

First, geometry information alone is not sufficient to determine the boundary of the resultant solid. In homogeneous object modeling, face direction is sufficient to eliminate the ambiguity for the boundary evaluation (Fig. 1). However, for a heterogeneous solid, even for the solids with the same geometric boundary classification, the final geometry may be different due to the different material compositions. For example, in Fig. 4a, point p in the left figure is *on* the (interior) boundary separating the two regions in the solid C $(mA \neq mB)$, while point p in the right figure is *in* the interior of solid C $(mA = mB)$. For the situations where there is no "on/on" ambiguity, material semantics still complicates the geometric boundary classification. For example, in Fig. 4b, the point p is completely in the interior of solid B, but it should appear in the final solid C of the left figure,

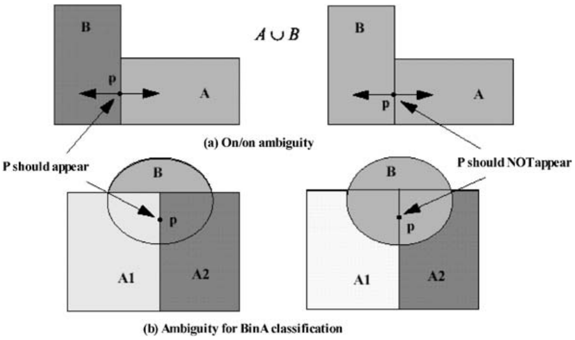


Fig. 4. Geometry information alone is not sufficient for boundary classification. (a) *on/on* ambiguity and (b) ambiguity for BinA classification.

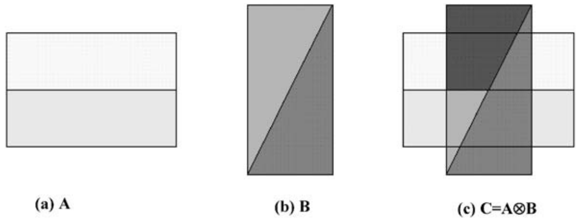


Fig. 5. Material region formulation remains unclear even after the boundary classification: (a) A, (b) B, and (c) $C = A \otimes B$.

and should not appear in the final boundary of solid C of the right figure.

In addition to geometric boundary classification, heterogeneous object modeling requires *material region forming* to complete the heterogeneous object representation. During the modeling process, the geometry and its partition are dynamically changing and so is the material composition at each partitioned region. Therefore, the lumps and shells that represent the regions need to be reorganized after each boundary classification. How to keep track of the region material information during the modeling process is an issue to be solved. Current *B*-Rep modeling systems do not directly support the region attributes propagation during the region merging/splitting processes.

For example, in Fig. 5, there are two regions in A and two regions in B. There are 12 lumps in the final solid C. The task of grouping the lumps into their respective regions (seven regions) and associating each region with the corresponding material function is referred to in this paper as *material region forming*.

In this paper, both the geometric boundary evaluation and material region forming are conducted based on a common computational framework: *direct face neighborhood alteration*.

In homogeneous set operations, the resultant solid boundary is a collection of classified boundary (Table 1). This classification is based on SMC augmented with the neighborhood information. In cellular object modeling, the geometric operation and volumetric attributes propagation are sequential. In heterogeneous object modeling, the steps involved in set operations in heterogeneous objects are the following. First, the SMC method is enhanced due to the usage of topological properties of heterogeneous objects. The face's two-sided neighborhood is altered according to material semantics in each region and faces' classification value. The boundary evaluation and material region forming are based on the altered neighborhood. In the following two sections of this paper, we detail the direct face neighborhood alteration and the enhanced SMC.

4. Direct face neighborhood operation

Neighborhood is a well-known concept from topology [25]. In heterogeneous object, since each face has two regions, we perceive the 3D face's neighborhood as a *two-sided face neighborhood* and represent it as a combination of two one-sided face neighborhoods from each adjacent region.

4.1. One-sided face neighborhood representation

The face neighborhood in each region is represented as a combination of normal direction of the face and material function of the region. Suppose point p lies on a face of region A, its neighborhood is represented as

$$nF_A = (dir\,A, mA). \qquad (1)$$

Here the $dir\,A$ is the region A's inward normal direction at point p; mA is the material composition function in region A.

For example, in Fig. 6a, the point p in region A's neighborhood is $nF(p) = (-n, mA)$.

4.2. Two-sided face neighborhood representation

Before we define a two-sided face neighborhood, we first define the neighborhood of the complement set of an object to ensure that each face has a two-sided face neighborhood.

Denote W, $W \subset E^3$, as the universal set. The *complement set* of a heterogeneous object S is defined as $S^c = W - S$. This complement set is also named as *NULL material region* since it does not contain any material substance. Due to the inclusion of S^c, each face in object S has two adjacent regions and they are either an *R-m* set in S or the NULL material region. Note a face neighborhood in NULL material region S^c is represented as $nF_{S^c} = (dir\,S^c, nil)$.

Denote a face's preserved reference normal direction at point p as n. The front side refers to the side of a face,

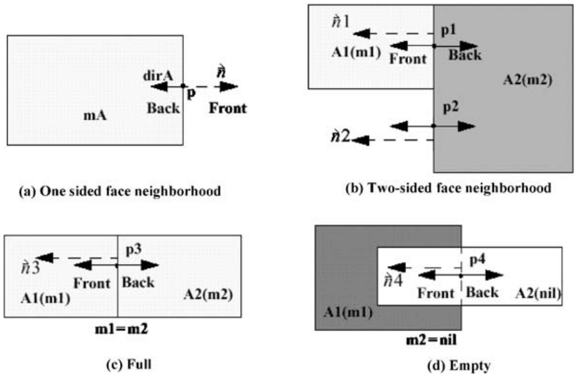


Fig. 6. Face neighborhood representation for a heterogeneous solid: (a) one-sided face neighborhood; (b) two-sided face neighborhood; (c) full; and (d) empty.

which is in front of p along the normal direction n. The opposite side is called a back side. So each face has two one-sided neighborhoods respectively in two adjacent regions, i.e., $nF_{front} = (RefNormal, m_{front})$, and $nF_{back} = (-RefNormal, m_{back})$.

A 3D face's complete neighborhood representation at point p is a combination of nF_{Front} and nF_{Back}.

$$NF(p) = nF_{Front} \,|\, nF_{Back}.$$

So the 3D face's neighborhood is a quadruple

$$NF(p) = (Ref\ Normal, m_{front})|(-Ref\ Normal, m_{back}). \tag{2}$$

From Eq. (2), we have the following interpretation of neighborhood concepts: *Empty* and *Full*.

$$Empty \equiv (n, nil)|(-n, nil), \tag{3}$$

$$Full \equiv (n, m)|(-n, m). \tag{4}$$

That is, when both sides of a face have null material, the neighborhood is empty and the face is in the exterior of the object. When both sides of a face have the same material function, the neighborhood is full and the face is in the interior of a region. During the regularization process, faces with either *empty* or *full* neighborhood shall be discarded.

For example, in Fig. 6b, the two-sided face neighborhoods of the points, $p1$ and $p2$, are $NF(p1) = (n1, m1)|(-n1, m2)$, $NF(p2) = (n2, nil)|(-n2, m2)$. In Fig. 6c, the point $p3$ has neighborhood $NF(p3) = (n3, m)|(-n3, m)$. Therefore, $p3$'s neighborhood is *full* and is completely interior to region B. In Fig. 6d, the point $p4$ lies on the boundary of $(A1–A2)$. So its neighborhood after the operation $(A1–A2)$ is $NF(p4) = (n4, nil)|(-n4, nil)$ and is *empty*.

4.3. Neighborhood operations

During the object construction process, i.e., $C = A \otimes B$, the face neighborhood NF alters according to the operation type "$\otimes$", and material semantics in A and B. This section details how NF alters according to the face classifications between A and B.

Let A and B be the collections of regions in heterogeneous objects, i.e. $A = \{A_1|^* A_2|^* \ldots|^* A_m\}$ and $B = \{B_1|^* B_2|^* \ldots|^* B_n\}$. Given the objects A and B, the faces from A and B, F_A and F_B, can be classified against each other. There are five types of SMC values: F_A in B, F_A out B, F_A on B/F_B on A, F_B in A, F_B out A (Fig. 7). With the inclusion of object complement set, F_A out B and F_B out A are equivalent to F_A in B^c, and F_B in A^c. Note, the faces F_A and F_B refer to the face sets in A and B after the intersection and sub-dividing.

During the intersection of the objects, face neighborhood NF changes when each face's NF interacts with another region or another face. This alteration can be illustrated according to the SMC. Therefore, corresponding to the five SMC values, there are five NF operations for the operation $A \otimes B$.

- $NF_A \otimes B_j$ for F_A inside region B_j.
- $A_i \otimes NF_B$ for F_B inside region A_i.
- $NF_A \otimes NF_B$ for F_A and F_B that are co-faces.
- $NF_A \otimes B^C$ for F_A outside the object B, i.e., F_A interacts with region B^C.
- $A^C \otimes NF_B$ for F_B outside the object A, i.e., F_B interacts with region A^C.

Fig. 7 shows the five neighborhood operations. Since different regions have different material operation semantics, the NF operations are carried out by combining two separate nF operations, each of which operates according to the residing region's semantics.

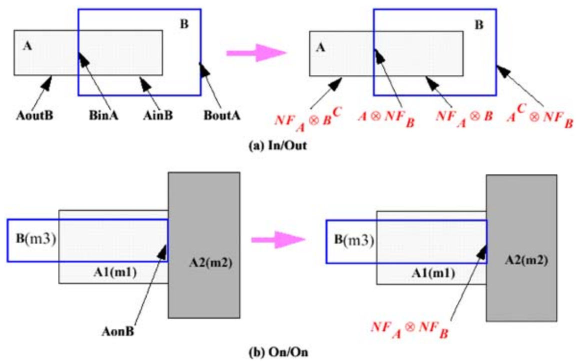


Fig. 7. Face membership classification and neighborhood operation: (a) in/out and (b) on/on.

4.3.1. F_A in region B_j

F_A's neighborhood operation with region B_j can be represented as

$$NF_A \otimes B_j = (nF_{AFront} \otimes B_j)|(nF_{ABack} \otimes B_j).$$

Here nF_{AFront} and nF_{ABack} refer to the face F_A's front region and back region's neighborhood. One-sided face neighborhood in region A_i is referred to as nF_{A_i}. The face neighborhood for the object A's complement set A^c is noted as nF_{A^c}.

An example of F_A interacting with region B is shown in Fig. 8 (bold line). From the four cases in the union operation, we have the following neighborhood alteration rules:

$$nF_{A_i} \cup B_j = \begin{cases} nF_{A_i}, & mA = mB, \\ nF_{A_i}, & pA > pB, \\ (dir\ A_i, mB), & pA < pB, \\ (dir\ A_i, mA \oplus mB), & pA = pB. \end{cases} \quad (5)$$

The face's NULL neighborhood operation can be represented by the following equations:

$$nF_{A^c} \cup B = (dir\ A^c, mB). \quad (6)$$

For the subtraction operation, we have

$$nF_{A_i} - B = (dir\ A_i, nil), \quad (7)$$

$$nF_{A^c} - B = (dir\ A^c, nil). \quad (8)$$

For the partition operation, we have the similar semantics derivation as union operation.

Note, if any object has more than one region, the one-sided face neighborhood operations are conducted separately. For example, in Fig. 9, the two-sided face neighborhood operation at point p is $NF(p) = (nF_{A_1} \otimes B)|(nF_{A_2} \otimes B)$. Each of the one-sided face neighborhood operation follows the same semantics defined above.

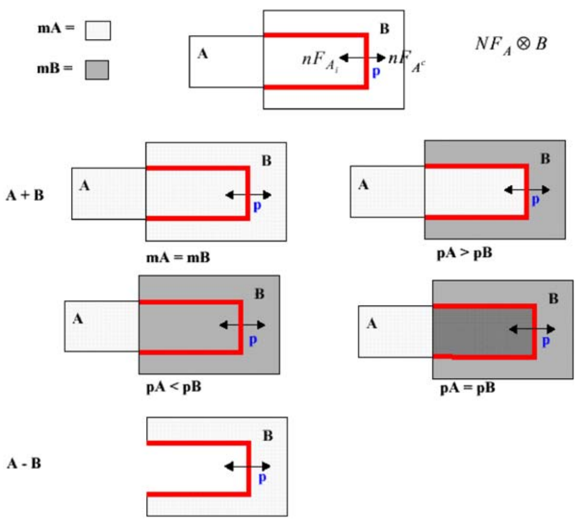


Fig. 8. Neighborhood operations for FA in B.

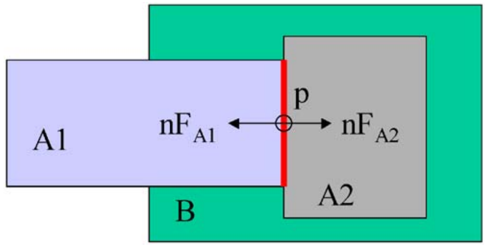


Fig. 9. Neighborhood operation in multi-region.

The complete neighborhood operations for F_A in region B_j are listed in Table 2.

4.3.2. F_B in region A_i

Similar to F_A's neighborhood in region B_j, we have the following neighborhood operation for F_B in A_i.

$$A_i \otimes NF_B = (A_i \otimes nF_{BFront})|(A_i \otimes nF_{BBack}). \quad (9)$$

The one-sided face neighborhood operation for F_B in region A_i is also listed in Table 2.

4.3.3. F_A on B/F_B on A

When F_A and F_B are co-faces, the neighborhood operation for F_A and F_B is $NF_A \otimes NF_B$. So we have the following operation representation:

$$NF_A \otimes NF_B = (nF_{AFront}|nF_{ABack}) \otimes (nF_{BFront}|nF_{BBack}). \quad (10)$$

$$NF_A \otimes NF_B = \begin{cases} (nF_{AFront} \otimes nF_{BFront})|(nF_{ABack} \otimes nF_{BBack}) \\ \qquad \text{if } dir(Fa) = dir(Fb), \\ (nF_{AFront} \otimes nF_{BBack})|(nF_{ABack} \otimes nF_{BFront}) \\ \qquad \text{if } dir(Fa) = -dir(Fb). \end{cases} \quad (11)$$

The one-sided face neighborhoods nF operate with each other only when their inward directions are the same. It is not applicable for the neighborhoods with different inward directions. For example, in Fig. 10, faces within object B are only involved with the operations on A_1, not A_2.

Referring to Fig. 10, we have the following neighborhood alteration rules for co-face situations:

$$nF_{A_i} \cup nB_j = \begin{cases} nF_{A_i}, & mA = mB, \\ nF_{A_i}, & pA > pB, \\ nF_{B_j}, & pA < pB, \\ (dir A_i, mA \oplus mB), & pA = pB. \end{cases} \quad (12)$$

The face's NULL neighborhood operation can be represented by the following equations: $nF_{A_i} \cup nF_{B^c} = nF_{A_i}$ and $nF_{A^c} \cup nF_{B_j} = nF_{B_j}$.

Table 2
One-sided face neighborhood operations

$A\otimes B$		$F_A \text{in} B$		$F_B \text{in} A$		$F_A \text{on} B$ or $F_B \text{on} A$	
		$nF_{A_i}\otimes B_j$	$nF_{A^c}\otimes B_j$	$A_i\otimes nF_{B_j}$	$A_i\otimes nF_{B^c}$	$nF_{A_i}\otimes nF_{B_j}$	$nF_{A_i}\otimes nF_{B^c}$
Additive	$mA=mB$	nF_{A_i}	$(dir\ A^c, mB)$	nF_{B_j}	$(dir\ B^c, mA)$	nF_{A_i}	
	$pA>pB$	nF_{A_i}		$(dir\ B, mA)$		nF_{A_i}	nF_{A_i}
	$pA<PB$	$(dir\ A, mB)$		nF_{B_j}		nF_{B_j}	
	$pA=pB$	$(dir\ A, mA\oplus mB)$		$(dir\ A, mA\oplus mB)$		$(dir\ A, mA\oplus mB)$	
Subtractive		nil	nil	nil	$(dir\ B^c, mA)$	nil	nF_{A_i}
Partition	$mA=mB$	nF_{A_i}	nil	nF_{B_j}	$(dir\ B^c, mA)$	nF_{A_i}	nF_{A_i}
	$pA>pB$	nF_{A_i}		$(dir\ B, mA)$		nF_{A_i}	
	$pA<PB$	$(dir\ A, mB)$		nF_{B_j}		nF_{B_j}	
	$pA=pB$	$(dir\ A, mA\oplus mB)$		$(dir\ A, mA\oplus mB)$		$(dir\ A, mA\oplus mB)$	
$A\otimes B$		$F_A \text{on} B$ or $F_B \text{on} A$		$F_A \text{out} B$		$F_B \text{out} A$	
		$nF_{A^c}\otimes nF_{B_j}$	$nF_{A^c}\otimes nF_{B^c}$	$nF_{A_i}\otimes B^c$	$nF_{A^c}\otimes B^c$	$A^c\otimes nF_{B_j}$	$A^c\otimes nF_{B^c}$
Additive	$mA=mB$	nF_{B_j}	nil	nF_{A_i}	nil	nF_{B_j}	nil
	$pA>pB$						
	$pA<PB$						
	$pA=pB$						
Subtractive		nil	nil	nF_{A_i}	nil	nil	nil
Partition	$mA=mB$	nil	nil	nF_{A_i}	nil	nil	nil
	$pA>pB$						
	$pA<PB$						
	$pA=pB$						

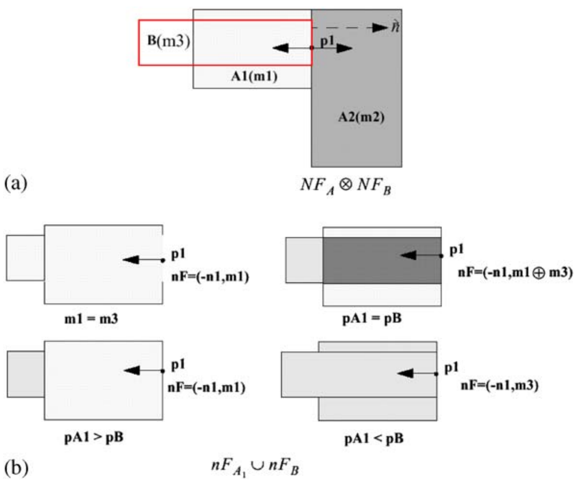


Fig. 10. Neighborhood operations for 3D faces (*on*/*on*): (a) $NF_A\otimes NF_B$ and (b) $nF_A\cup nF_B$.

Fig. 10 only lists the cases for union operations. For subtraction and partition operations, material neighborhood alteration rules can be derived similarly (shown in Table 2).

4.3.4. F_A out B

We have the following equations for neighborhood operations for F_A outside B:

$$NF_A\otimes B^c = (nF_{AFront}\otimes B^c)|(nF_{ABack}\otimes B^c). \quad (13)$$

Since F_A is outside of B, its neighborhood is not affected by region B regardless of the operation type, so $nF_{A_i}\otimes B^c = nF_{A_i}$ and $nF_{A^c}\otimes B^c = nF_{A^{c0}}$.

4.3.5. F_B out A

The equations for neighborhood operations for F_B outside A:

$$A^c\otimes NF_B = (A^c\otimes nF_{BFront})|(A^c\otimes nF_{BBack}). \quad (14)$$

When F_B is outside A, the material in region A has no effect on F_B's NULL neighborhood nF_{B^c}, i.e. $A^c\otimes nF_{B^c} = nF_{B^c}$ However, the operation type "$\otimes$" affects the resultant nF_{B_j}, i.e. $A^c\cup nF_{B_j} = nF_{B_j}$, $A^c - nF_{B_j} = nil$ and $A^c/nF_{B_j} = nil$.

The complete list of neighborhood operations is shown in Table 2. These neighborhood operations are consistent with material region semantics presented in Fig. 3. Fig. 3 illustrates the neighborhood alteration of a point p under all kinds of operations. Point p lies on the face of object A. Before the operations, it has neighborhood $NF(p) = (n, mA)|(-n, nil)$. After the operations, $NF(p)$ exhibits different values for different operations, such as $(n, mA)|(-n, mA) \equiv Full$, $(n, mB)|(-n, mB) \equiv Full$, $(n, mB)|(-n, mA)$, $(n, mA \oplus mB)|(-n, mB)$, $(n, nil)|(-n, nil) \equiv Empty$, $(n, mA)|(-n, nil)$, $(n, mB)|(-n, nil)$ *and* $(n, -n, nil)$, $(n, mB)|(-n, nil)$ *and* $(n, mA \oplus mB)|(-n, nil)$.

4.4. Algorithm (neighborhood operation algorithm)

Based on these defined neighborhood operations, for any heterogeneous object operation $A \otimes B$, we have the following neighborhood processing algorithm to calculate the face neighborhood change (Fig. 11). Note, in this algorithm, we assume that the face membership classification is given.

First, each face's one-sided neighborhoods, nF_{Front} and nF_{Back}, are found. Then according to the operation type and face classification, face neighborhood operations are performed, respectively, from Eqs. (5), (9), (10), (13), and (14). Each of the NF operations can be further decomposed into two nF operations as listed in Table 2.

Neighborhood_Operation Algorithm (A, B, ⊗)
Input: Object A, Object B, operator ⊗
Output: Neighborhood of each face in A and B

```
{
      For each face F_A in A { //after classification and
         subdividing faces
        GetNeighborhood NF_A = nf_AFront |nf_ABack ;
        GetClassification( F_A, B);
        Case FA in B:
             NF_A = NF_A ⊗ B ;
        Case FA on B:
        //keep NFA the same since it is processed in F_BonA
case;
        Case FA out B:
             NF_A = NF_A ⊗ B^c
      } // Endof FA in A
      For each face F_B in B {
        GetNeighborhood NF_B = nfBFront|nfBBack ;
        GetClassification( FB , B)
        Case F_B in A:
             NF_B = A ⊗ NF_B ;
        Case F_B on A:
             NF_B = NF_A ⊗ NF_B ;
        Case F_B out A:
             NF_B = A^c ⊗ NF_B ;
      } // Endof FB in B
}
```

Fig. 11. Neighborhood operation algorithm.

4.5. Boundary evaluation for heterogeneous objects

The final geometric boundary of constructive operations can be derived from the direct face neighborhood processing. It is described by the following lemma.

Lemma 1. *A face remains in the resultant solid if and only if the face's two adjacent regions have different material composition functions* (*Boundary Evaluation*).

The lemma suggests:

1. For any face F, if NF is full, i.e. $NF = (n, m)|(-n, m)$, or NF is empty, i.e. $NF = (n, nil)|(-n, nil)$, then the face F shall be removed.
2. For any edge E, if E only has two adjacent faces and they are co-faces, then the E shall be removed.

Once face neighborhood has been properly processed, the edge-classification can be easily derived. For the edges that have only two adjacent faces, and both are co-faces, then the edges shall be eliminated.

Fig. 12 shows an example, in which objects A and B each has two regions with different material semantics. After the neighborhood operations, the result is shown in Fig. 12b. This boundary evaluation algorithm then "regularizes" the model according to the lemma for boundary evaluation. Faces, such as $F1$ and $F2$, are eliminated after the evaluation in Fig. 12c.

The boundary evaluation algorithm is shown in Fig. 13.

4.6. Material region forming

In our representation, each face has references to the residing regions' material functions. To maintain a correct relationship between the face and its adjacent region's material semantics, material information processing is necessary before the new shells and lumps are organized. Direct face neighborhood operations solve this task efficiently. After the neighborhood alteration, each face carries proper material information. Regions

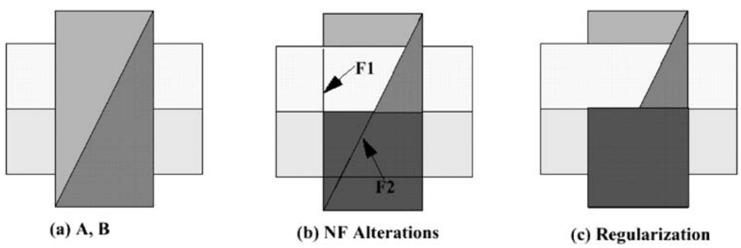


Fig. 12. Neighborhood-based boundary evaluation: (a) A, B; (b) NF alterations; and (c) regularization.

```
Boundary_Evaluation Algorithm (A, B, ⊗)
Input: Object A, Object B, and operator ⊗
Output: Boundary portions in A ⊗ B
{
Intersect A with B;
Classify F_A against B;
Classify F_B against A;
Neighborhood_Operation( A, B, ⊗);
For each face F in A and B{
        Get the neighborhood NF=(nF_Front|nF_Back);
        if (nF_Front->material==nF_Back->material){
                eliminate face F;
                }
        }
//eliminate edge E if it separates co-faces
For each edge E {
        if((E has only two adj faces, F1 and F2) && co-face(F1,F2))
                eliminate edge E;
        }
}
```

Fig. 13. Boundary evaluation algorithm.

```
Material Region Forming Algorithm
Input: Object A, Object B, and operator ⊗
Output: R-m sets in A ⊗ B
{
Boundary Classification(A, B,⊗);
Get_all_the_material_functions (M, nM);//nM is the number of R-m sets
For (i=0;i<nM;i++){
     For each face F in A and B{
        Get the neighborhood NF(nF_Front, nF_Back);
        if (nF_Front->material == M[i] || nF_back->material == M[i]) {
                If ( F∩R_mset[i]≠∅ )
                    Add the F into R-mset[i];
        }
     }
}
```

Fig. 14. Material region forming algorithm.

can be easily formed and are associated with material functions according to the following lemma.

Lemma 2. *A collection of faces forms a region if and only if all the faces are topologically connected and share one material composition function M (material region forming).*

$$R(m) = \{F_i \,|\, \exists F_j \subset R | (F_i \cap F_j \neq \varnothing);\ \ nF_i = (n, m)\}. \quad (15)$$

The material region classification algorithm is shown in Fig. 14.

5. SMC for heterogeneous objects

During the Boolean operations, the face neighborhood changes according to the SMC. Therefore, in order to alter the face neighborhood in accordance with the region material semantics, each face's SMC related to the other objects needs to be known.

Many methods have been proposed for SMC [21]. However, in this paper, we enhance the existing methods for SMC by utilizing the unique characteristics of the internal boundary of heterogeneous objects to eliminate the unnecessary complex geometric intersection computation. This method automatically infers the SMC according to the topological relationship. In addition to the inference, it also propagates SMC value; i.e., some portions' SMC value can be derived by propagation if the adjacent portion's SMC is known.

5.1. Theoretical basis

Intersection loop, $IL(\alpha, \beta)$, refers to the loops formed due to the intersection of two objects/regions α and β. It is a collection of edges that are shared by both objects/regions α and β.

Let E be the collection of the edges in the resultant solid from the intersection between α and β. We have $IL(\alpha, \beta) = \{x \,|\, x \in E, x \in \alpha, x \in \beta\}$.

There are two types of interaction loops, the interaction loop between the objects and the interaction loop between the regions. The first type is an object interaction loop, noted as IL_1, consisting of intersecting edges from exterior boundaries of the two objects. The second type is a region interaction loop, noted as IL_2, and it can be composed of edges from interior boundaries.

For example, in Fig. 15, object A and object $B = \{B1|^* B2\}$ intersect with each other. Fig. 15a and b show IL_1 and IL_2. The region interaction loops, $L1$ and $L2$ include the interaction between the internal boundaries of object B with another object/region A. These internal boundaries do not appear in the object interaction loop (L). So the collection of IL_1 is a subset of IL_2 collection, i.e., $\prod(IL_1) \subseteq \prod(IL_2)$. For example, in Fig. 15 edge $e0$ appears in IL_2, but not in IL_1.

Depending on the edge's position in IL_2 relative to the boundary S of the object, the IL_2 can have edges that are interior to the object, and the edges that lie on the boundary of the object. We call these edges IL_2's "*in* edges" and "*on* edges" respectively. For the IL_2's *in* edges, $IL_{2,in} = \{x \,|\, x \in IL_2, x \text{ in } S\}$. For the region IL_2's *on* edges, $IL_{2,on} = \{x \,|\, x \in IL_2,\ x \text{ on } S\}$ (Fig. 15b). Note, all the IL_1 are on the object boundary.

The interaction loops partition the object boundaries into several sections. Each section is called a "portion" in this paper. Let *portion* refer to the collection of connected faces bounded by IL with no internal edges from other ILs. Mathematically, $P(IL_i) = \{x \,|\, x \in F, \sum F \supseteq IL_i, \forall e \in IL_j, e \notin F\}$, where e is an edge of the interaction loop IL.

In Fig. 15b, there are three interaction loops: one object IL_1 (L) and two region IL_2 ($L1$ and $L2$). Correspondingly, these three loops partition the object A into different portions as shown in Fig. 15c. For example, the loop $L1$ partitions the object A into three portions, $P4, P6, P7$. Note, $P6 \cup P7$ does not

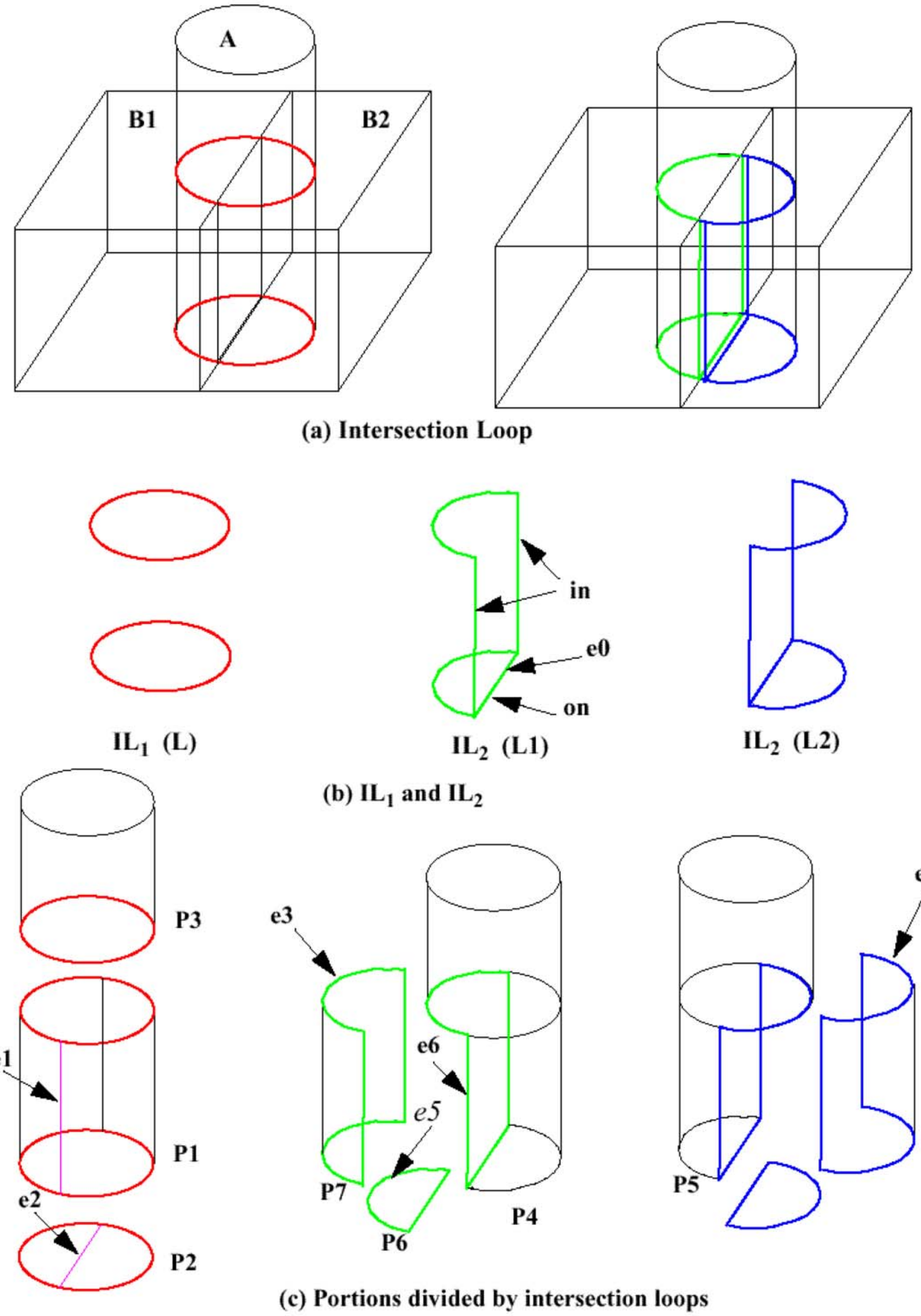


Fig. 15. Intersection loop and portion: (a) intersection loop; (b) IL_1 and IL_2 and (c) portions divided by intersection loops.

form a portion since otherwise there would be an internal edge $e5$ in the portion. For illustration purpose, portion formation in Fig. 15c is separate for each interaction loop. The actual portion formation in the later algorithm is shown in Fig. 19b.

An interaction loop's *adjacent regions* (AR) are the regions on which the edges of the IL lie. Interaction loop's adjacent regions within a solid S is noted as $AR(IL, S)$. Mathematically, $AR(IL, S) = \{x \mid x \in R, x \in S, \forall e \in IL, e \in R_i\}$, where R is the collection of regions and S is the solid. Suppose $L = IL(A, B)$, then $AR(L, B)$ does not include any region in A.

Different IL_2 may share common edges. An IL_2's *sibling interaction loop* refers to the another IL_2 with which common edges are shared.

In Fig. 15, the loop $L1$ (IL_2) has the sibling loop $L2$ (IL_2) and vice versa. $L1$'s and $L2$'s adjacent regions include $\{B1, B2, A, NULL\}$. $NULL$ is the complement set of the object.

5.1.1. Properties of portion membership classification for heterogeneous objects

Suppose candidate set α is classified against reference set β. The SMC function of any geometric entity x within α against β is represented as

$$SMC(x, \beta) = \begin{cases} in, & x \in \beta, x \cap \partial\beta = 0, \\ on, & x \in \beta, x \cap \partial\beta \neq 0, \\ out, & x \notin \beta, \end{cases}$$

where $\partial\beta$ refers to β's boundary. With the above definitions, we have the following propositions for SMC.

Proposition 1. *All the topological entities in a portion bounded by an interaction loop have one SMC value*: *in*, *on*, or *out*. Mathematically,

$$(\forall x, \forall y \subset P(IL)) \in \alpha \Rightarrow SMC(x, \beta) = SMC(y, \beta) = \begin{cases} in \\ on \\ out \end{cases}$$

Proposition 2. *If any portion bounded by an object interaction loop contains the region $IL_{2,in}$, then that portion's SMC value relative to the object is "in" (object In).*

$$P(IL_1) \supseteq IL_{2,in} \Rightarrow SMC(P(IL_1), \beta) = in.$$

Proof. According to our definition, IL_2's "in" edges lie in the interior of the object. So these "in" edges have the SMC value "in", relative to the object. According to Proposition 1, the entire portion bounded by IL_1 that contains the IL_2 "in" edges has SMC value "in".

For example, in Fig. 15b and c, portion 1 contains edge $e1$, which is $IL_{2,in}$, so portion 1 from object A has the SMC value "*in*" relative to the object B. □

Proposition 3. *If any portion bounded by an object IL_1 contains the region $IL_{2,on}$, and the region IL_2 does not completely belong to IL_1, then that portion's SMC value is "on" (object On).*

$$P(IL_1) \supseteq IL_{2,on}, IL_{2,on} \not\subset IL_1 \Rightarrow SMC(P(IL_1), \beta) = on.$$

The proof of Proposition 3 is similar to the proof of Proposition 2.

For example, in Fig. 15b and c, portion 2 contains edge $e2$, which is $IL_{2,on}$, so portion 2 from object A has the SMC value "*on*" relative to the object B.

Note, any region $IL_{2,on}$ edges that also belong to IL_1 do not provide any information for SMC.

Proposition 4. *The intersection of a set of regions' outer portions has the SMC value "out" relative to the union of the set of regions (object Out).*

$$SMC(P(IL_{2_i}), R_i) = out \Rightarrow SMC\left(\bigcap_{i=1}^{n} P(IL_{2_i}), \bigcup_i R_i\right) = out.$$

This proposition's proof is straightforward according to the set operation property.

This proposition is especially useful for deducing a portion bounded by object IL_1, lying outside of an object. For example, in Fig. 15, portion 3 is outside region $B1$ and region $B2$. According to Proposition 4, portion 3 is outside of region $B = \{B1|^{*}B2\}$.

Proposition 5. *Any portion bounded by a region IL_2, if this portion satisfies the following conditions, the candidate region's SMC value is "in" (region in):*

(1) it does not contain any edges from other interaction loop;
(2) its boundary has such two edges that the only common adjacent region in the reference object Γ is the reference region itself.

Illustrative proof for Proposition 5

$$\left.\begin{array}{c} L = IL(\alpha, \beta), \beta \subseteq \Gamma \\ \exists E_1, E_2 \in L, \ s.t. \ AR(E_1, \Gamma) \cap AR(E_2, \Gamma) = \{\beta\} \end{array}\right\} \Rightarrow SMC(P(L), \beta) = in.$$

Proof. An edge in the interaction loop is shared by the faces from both the reference object and the candidate object. In heterogeneous objects, each face has two adjacent regions. Therefore, any edge in the interaction loop has at least two adjacent regions in the reference object.

Assume $AR(E_1, \Gamma) = \{\beta, R_1', \ldots\}$, $AR(E_2, \Gamma) = \{\beta, R_2', \ldots\}$ (Fig. 16). Since $AR(E_1, \Gamma) \cap AR(E_2, \Gamma) = \{\beta\}$, we have $\{R_1', \ldots\} \cap \{R_2', \ldots\} = \emptyset$.

Let $x1$ and $x2$ be points, respectively, from $E1$ and $E2$. Get the neighborhood ball of $x1$, $x2$ as $N1$, $N2$. $N1$ consists of volume from $\{\beta, R_1', \ldots\}$ and $N2$ consists of volume from $\{\beta, R_2', \ldots\}$.

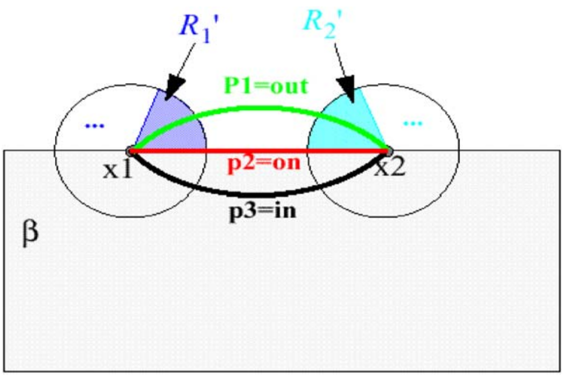


Fig. 16. Illustrative proof for Proposition 5.

The portion is bounded by $L = IL(\alpha, \beta)$. According to Proposition 1, the portion $P(L)$, relative to the reference region β, has one SMC value, either *in*, *on*, or *out*.

Assume $P(L)$ from α is *out* β, as shown in Fig. 16(P1). Since the portion is a connected set of faces, it has to go through $\{R_1', \ldots\}$ and $\{R_2', \ldots\}$ so that the portion is outside β. $\{R_1', \ldots\} \cap \{R_2', \ldots\} = \varnothing$. Therefore, the portion has to be across the boundaries of $\{R_1', \ldots\}$ and $\{R_2', \ldots\}$. That is, there must be some interaction edges within the portion P. This is contradictory to condition 1 (*no edges from other IL*), so the portion cannot be *out*.

Assume $P(L)$ from α is *on* β, as shown in Fig. 16(P2). For the similar reason as above, the portion $P(L)$ that connects $x1$ to $x2$ must have edges that go across $\{R_1', \ldots\}$ and $\{R_2', \ldots\}$. Since faces in α and β are on, these crossing edges also lie in α. Therefore they are also part of IL. This again is contradictory to the condition 1 (*no edges from other IL*), so this portion cannot be *on*.

Therefore, $P(L)$ from α is *in* β. □

For example, in Fig. 15, portion 3 contains edges $e3$ and $e4$. In the reference set B, $AR(e3, B) = \{B1, NULL\}$, $AR(e4, B) = \{B2, NULL\}$. So we have portion 3 in region $NULL$, i.e., portion 3 is outside of B.

For another example, consider portion 4 in Fig. 15. Portion 4 contains edges $e3$ and $e6$. $AR(e3, B) = \{B1, NULL\}$, $AR(e6, B) = \{B1, B2\}$. So $AR(e_3, B) \cap AR(e_6, B) = \{B_1\}$. This portion satisfies condition 2, but it does not satisfy condition 1. Portion 4 contains edge $e4$, so it is not in region $B1$.

If we consider the null material region as a separate region, then Proposition 4 is a special case of Proposition 5 with β representing the null material region.

Proposition 6. *If a portion from the candidate set* α *is bounded by* $IL_2 = IL(\alpha, \beta_1)$ *and it contains an edge from a sibling region* β_2*'s* $IL_2 = IL(\alpha, \beta_2)$*, then, relative to the reference region* β_1*, that portion has a SMC value "out" (region out).*

$$IL_2 = IL(\alpha, \beta_1), \exists e \in IL_{2_{sib}} = IL(\alpha, \beta_2), P(IL_2) \supset e \Rightarrow SMC(P(IL_2), \beta_1) = out.$$

Proof. According to the definition of sibling IL, we know region β_1's sibling region β_2 is outside of region β_1. Therefore, for any portion P in α formed by the loop $IL_2(\alpha, \beta_1)$, if the portion P contains the intersection edges from the sibling loop in β_2, then portion P is outside of region β_1. □

In Fig. 15, portion 4 contains edge $e4$, which is in the loop $L2$, the sibling loop of loop $L1$. Therefore, portion 4 bounded by loop $L1 = IL(A, B1)$ is outside of region $B1$. Likewise, portion 5 is outside of region $B2$.

The above propositions demonstrate that many portions' SMC values can be inferred without any geometric calculation, and these propositions give the conditions for the SMC inference.

5.1.2. SMC propagation

In addition to the above set (portion) membership classification propositions, there are other situations where, once a portion's SMC is known, the other portions' SMC can be inferred. We refer to such an inference process as an *SMC propagation process*.

Proposition 7. *Suppose P is a portion from region* α_1*, P against* β *is known. Region* α_1 *has adjacent regions* $\alpha_2, \alpha_3, \ldots, \alpha_n$ *that interact with* β*. Then any portion in* $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n\}$ *that shares topological entities with portion P has the same SMC value as P has (SMC Propagation) against* β*.*

This can be readily proved from Proposition 1.

Consider the same example in Fig. 15, in which B is classified against A, as shown in Fig. 17. Suppose by point membership classification, $B1$out$A$ is known. By SMC propagation, $B2$out$A$ can also be inferred since the two portions (left and right) share the faces, such as $f1$.

5.2. SMC algorithm

We can now formulate the outline of the set (portion) membership classification algorithm as follows (see Fig. 18). Each step is also illustrated in Fig. 19:

Step 1: Object interaction and intersection loop identification.

Intersection loops consist of edges from the reference object and the candidate object. These intersection loops are divided into two groups: object IL_1 and region IL_2. For the edges that are in IL_2 but not in IL_1, i.e., $IL_2 - IL_1$, they are divided into two groups: $IL_{2,in}$ and $IL_{2,on}$.

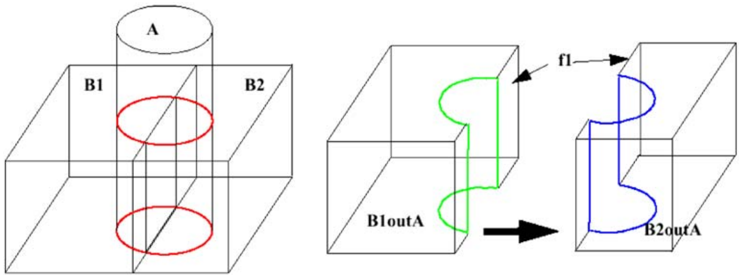


Fig. 17. SMC propagation.

Portion Membership Classification Algorithm

Input: Object A, R-m set B

Output: Boundary B's SMC against A

```
{
Interaction_loop_identification(A,B);
Portion_Formation;
Reasoning_Portions_IN;
Reasoning_Portions_ON;
Reasoning_Portions_OUT;
For each remaining unknown portion {
    Get_one_point_x_from_portion(x,p);
    do_Point_Membership_Classification(p,β);
    propagate_the_SMC(p,β);
    }
}
```

Fig. 18. Portion membership classification algorithm for heterogeneous objects.

Step 2: Portion formation.

Connected faces bounded by an IL form the portions. Fig. 19a and b show the intersection loops and portions.

Step 3: Reasoning for the "*in*" portions.

Check each portion's boundary. According to Proposition 4, if $AR(E_1, \Gamma) - AR(E_2, \Gamma) = \{\beta\}$, then the portion has SMC value "*in*".

Step 4: Reasoning for the "*on*" portions.

According to Proposition 3, if $IL_{2,on} - IL_1 \neq \varnothing$, all the portions that are bounded by IL_1 and contain the $IL_{2,on} - IL_1$ have the SMC value "*on*".

Step 5: Reasoning for the "*out*" portions.

Consider the null material region as the reference region β. If $AR(E_1, \Gamma) - AR(E_2, \Gamma) = \{\beta\}$, then the portions are in the reference region β, i.e. the null material region. Therefore the portions have the SMC value "*out*".

Fig. 19c–e show the portions that can be inferred without any geometric calculations.

Step 6: Point membership classification and SMC propagation.

For each of the remaining portions with an unknown SMC value, do a point membership classification. Suppose, by point membership classification, a portion α's SMC is known against β. Check all the adjacent regions of α. If any portion from these adjacent solids also interact with α, these portions have the same SMC value as α has.

Fig. 19 shows that by point membership classification $B1$out$A$ is known. By SMC propagation, $B2$out$A$ can also be inferred since two portions share the faces. Similarly, $B2$in$A$ can be deduced once the portion $B1$in$A$ is known.

6. Implementation

A prototype system for heterogeneous object design has been implemented based on ACIS [26] on a HP-UX 10.0 machine. The languages used were C++ and Scheme. After the parts are modeled through direct face neighborhood alteration according to the operations defined earlier, for display purposes, the parts were then decomposed into several regions with different material gradient information. The regions with their respective material information were transferred to our in-house software, Heterogeneous Solid Modeler (HSM) [27], for display.

Fig. 20 shows the sample part from Fig. 17. First, the parts are classified against each other by the SMC algorithm (Fig. 19). By direct face neighborhood alteration, the system gives different results, depending on the priority of each primitive. The bottom half of the figure is the shaded cross-section of the parts.

A cutting tool made of gradient alumina–aluminide alloys (3A) [28] is shown in Fig. 21. In this example, the part was constructed by two partition operations. The first partition operation replaced the ceramic with Al_2O_3. The second partition operation replaced the material composition around the shaft area with functionally gradient material 3A. Fig. 21 illustrates how the face neighborhood changes during the modeling process.

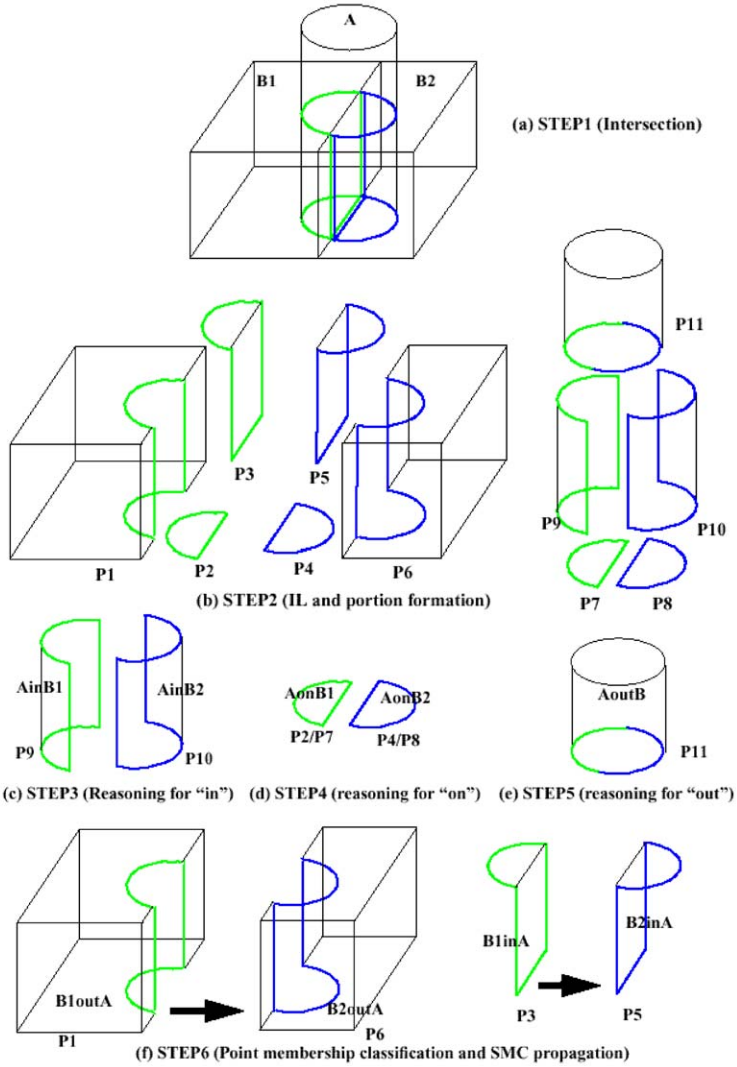


Fig. 19. Portion membership classification: (a) Step 1 (intersection); (b) Step 2 (*IL* and portion formation); (c) Step 3 (reasoning for "*in*"); (d) Step 4 (reasoning for "*on*"); (e) Step 5 (reasoning for "*out*") and (f) Step 6 (point membership classification and SMC propagation).

Fig. 22 shows an MEMS fabrication process modeled through the system. Two types of operations, additive and subtractive, are used. The color changes illustrate the face neighborhood changes during the modeling process. In the last step (Fig. 22e) electrode overrides acetone. So all the neighborhood of the faces from acetone are changed to electrode if they are "inside" electrode.

Note, in all the above examples, 3D regions are formed only at the last stage for the sake of material gradient display. These examples demonstrate that the direct face neighborhood alteration method is a feasible, effective, and efficient method for heterogeneous object modeling.

7. Discussion

7.1. Comparison with 3D cell-based cellular object modeling

Compared with many current cellular modeling systems [20,26], constructive operations based on face neighborhood alteration have three advantages: (1) it avoids unnecessary 3D cell/region formation; (2) it eliminates the radial-edge ordering; and (3) it utilizes the heterogeneous objects' topological characteristics to infer and propagate SMC.

Current cellular object modeling focuses on the geometric aspects. Due to the lack of physical informa-

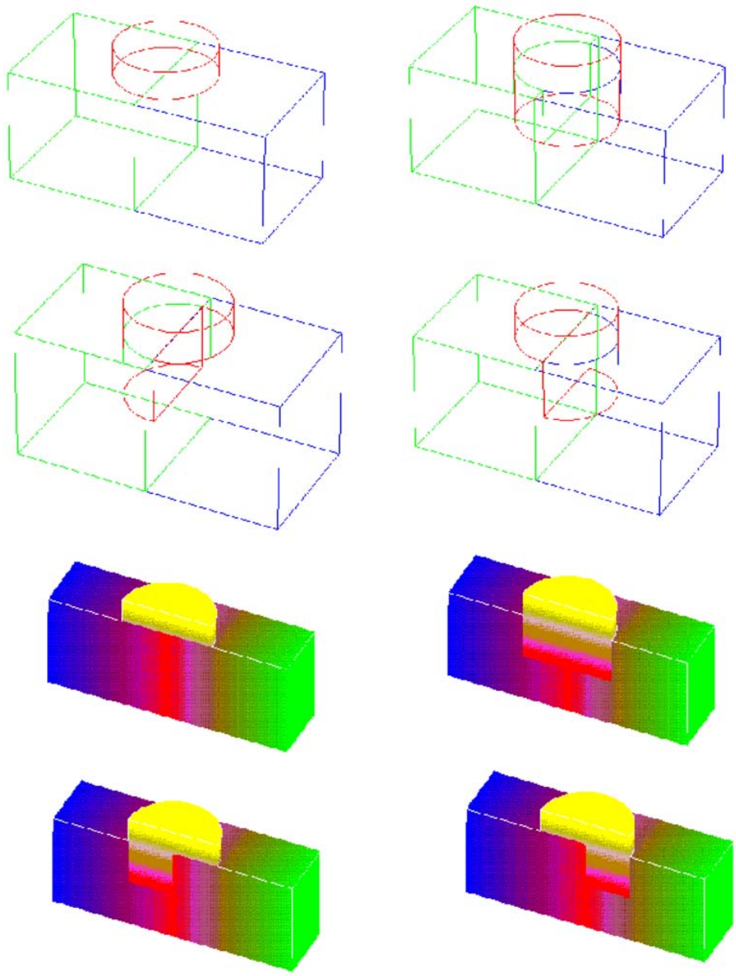


Fig. 20. Sample part for face neighborhood alteration.

tion, general cellular object modeling unnecessarily generates all the intersection regions. Suppose object A and B each consisted of m and n regions; therefore, there are possibly $(mn + m + n)$ 3D cells for the union operation. It is up to the application users to departition the over-segmented object. For example, in ACIS 3D cell-based modeling [26], cell computing is done by finding the nearest face from a seed face around each edge (radial-edge ordering). Volume attribute propagation algorithm follows. The 3D cells are then grouped together according to the volume attribute and are regularized.

In our two-sided face neighborhood algorithm, the fundamental difference from cellular object modeling is that material attributes are directly processed for each face/face and face/region interaction. Therefore, there is no need for intermediate 3D cell creation. The material region is formed only when it is necessary. It avoids the unnecessary 3D cell shell/region forming.

Therefore direct face neighborhood alteration is expected to be an efficient method for heterogeneous object modeling.

7.2. Persistent region naming in the heterogeneous objects

The face neighborhood operation method presented in this paper can also serve as a persistent naming scheme for region naming.

During the design process, the part topology changes. Consequently, all the attributes attached to the topology entities need to be correctly identified during the editing process. That is, each topological entity (vertex, edge, face) shall have a unique name. This is the persistent naming problem [29–31].

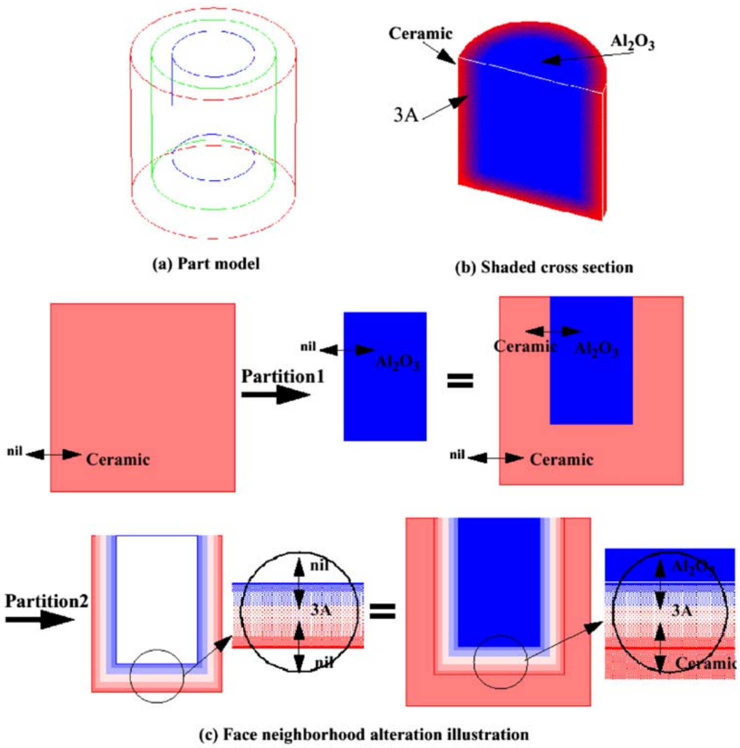


Fig. 21. Cutting tool. (a) Part model; (b) shaded cross-section; and (c) face neighborhood alteration illustration.

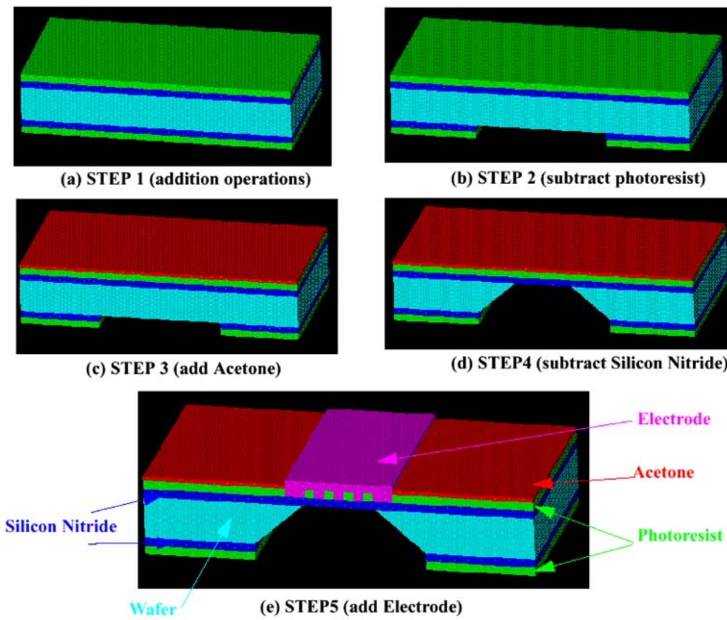
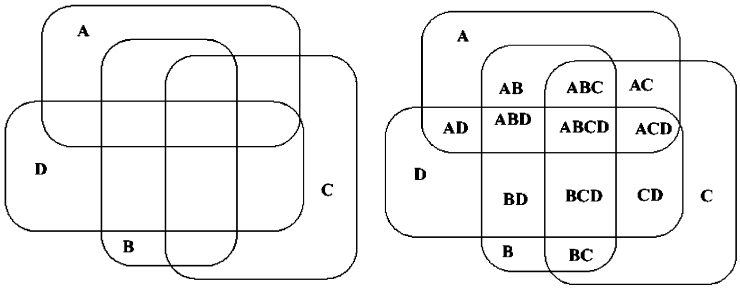
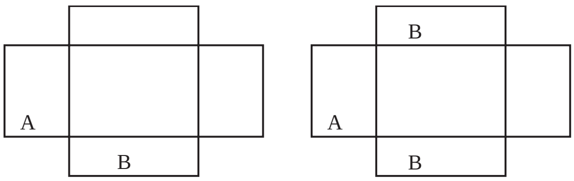


Fig. 22. MEMS fabrication process: (a) Step 1 (addition operations); (b) Step 2 subtract photoresist; (c) Step 3 (add acetone); (d) Step 4 (subtract silicon nitride); and (e) step 5 (add electrode).



(a) Initial regions (b) Region names after interaction

Fig. 23. Region naming. (a) Initial regions and (b) region names after interaction.



(a) Initial regions (b) After interaction

Fig. 24. Region naming. (a) Initial regions and (b) after interaction.

In the context of heterogeneous object modeling, there is a new issue: the *region naming*. A heterogeneous object is composed of different regions, each having a different material composition function. During the constructive/editing process, the lumps and shells, of which the regions are represented, have to be re-organized after each operations. How to persistently associate material composition function with the corresponding region is a region naming issue.

With little change, the neighborhood alteration algorithm can serve as a naming scheme for regions in heterogeneous objects. It involves the following steps: (1) each attending primitive (*R-m* set) has one unique name; (2) equal priority is assumed for the operation attending primitives. Therefore, the region names are concatenated together whenever regions interact with each other. After the neighborhood processing, each intersection region has one unique name. The region naming scheme derived from the two-sided face neighborhood generates unique names for each *R-m* sets without resorting to 3D cells or geometric calculation.

Fig. 23 gives an example of the region naming. This example is from [11]. All the faces that have the same attributes form one region. Clearly, each region has one unique name.

It should be noted that this neighborhood alteration-based naming scheme is dependent on the region's SMC. Therefore, it does not distinguish the sub-regions that have the same SMC value. For example, in Fig. 24, there are two sub-regions named '*B*'.

8. Conclusion

This paper presents a novel method, *direct face neighborhood operation*, for constructive operations in heterogeneous object design. Through the defined face neighborhood operations, this method enables the direct face neighborhood change according to face membership classification and region material semantics. It then performs part geometric boundary evaluation and region material forming after the face neighborhood alteration.

The algorithms in this paper also utilize the heterogeneous object model's topological characteristics to infer the SMC. They demonstrate that direct face neighborhood processing is an effective and computationally efficient method for heterogeneous object modeling. It allows for concurrent geometric and material operations as opposed to sequential operations in the existing methods.

This face neighborhood alteration-based heterogeneous object modeling is part of our overall research efforts on feature-based heterogeneous object design. Future work shall further extend the direct face neighborhood alteration in feature-based design for heterogeneous objects.

References

[1] Jackson T, Liu H, Patrikalakis NM, Sachs EM, Cima MJ. Modeling and designing functionally graded material components for fabrication with local composition control. Materials and Design 1999;20(2/3):63–75.

[2] Kumar V. Solid modeling and algorithms for heterogeneous objects, Ph.D. dissertation, Mechanical Engineering, University of Michigan, 1999.

[3] Qian X, Dutta D. Features in the layered manufacturing of heterogeneous objects. Symposium of Solid Freeform Fabrication, The University of Texas, Austin, August 1998, p. 689–96.

[4] Qian X, Dutta D. Feature methodologies for heterogeneous object realization. In: Gibson I, editor. Software

solutions for rapid prototyping. UK: Professional Engineering Publishing; 2002. p. 225–61 (Chapter 7).
[5] Qian X, Dutta D. Physics-based modeling for heterogeneous objects. ASME Transactions. Journal of Mechanical Design 2003;125(3):416–27.
[6] Hoffmann CM. Geometric and solid modeling: an introduction. Los Altos, CA: Morgan Kaufmann Publishers, Inc.; 1989.
[7] Requicha AAG. Representations for rigid solids: theory, methods and systems. Computing Survey 1980;12(4): 437–64.
[8] Weiler KJ. Topological structures for geometric modeling. Ph.D. dissertation, Rennselaer. New York: Polytechnic Institute; 1986.
[9] Pratt M. Synthesis of an optimal approach to form feature modeling. Proceedings of the ASME Conference on Computers in Engineering, San Franciso, August, 1988.
[10] Rossignac J, O'Connor M. SGC: a dimension-independent model for pointsets with internal structures and incomplete boundaries. In: Wozny M, Turner J, Periss K, editors. Geometric modeling for product engineering. IFIP, Amsterdam: Elsevier (North-Holland); 1990. p. 145–80.
[11] Rossignac J, Requicha AAG. Constructive non-regularized geometry. Computer-Aided Design 1991;23(1):21–32.
[12] Feito FR, Torres JC. Boundary representation of polyhedral heterogeneous solids in the context of a graphic object algebra. The Visual Computer 1997;13:64–77.
[13] Lienhardt P. Topological models for boundary representation: a comparison with n-dimensional generalized maps. Computer-Aided Design 1991;23(1):59–82.
[14] Middleditch AE, Reade CMP, Gomes AJ. Set-combinations of the mixed dimensional cellular objects of the Djinn API. Computer-Aided Design 1999;31:683–94.
[15] Kumar V, Dutta D. An approach to modeling and representation of heterogeneous objects. ASME Journal of Mechanical Design 1998;120(4):659–67.
[16] Park, Seok-Min, Crawford RH, Beaman JJ. Functionally gradient material representation by volumetric multi-texturing for solid freeform fabrication. 11th Annual Solid Freeform Fabrication Symposium, Austin, TX, August 7–9, 2000.
[17] Biswas A, Shapiro V, Tsukanov I. Heterogeneous material modeling with distance fields. Technical Report, University of Wisconsin-Madison, Mechanical Engineering Department, Spatial Automation Laboratory, SAL-2002-4.
[18] Wu Z, Soon SH, Feng L. NURBS-based volume modeling. International Workshop on Volume Graphics, Swansea, UK, 1999, p. 321–30.
[19] Mantyla M. Boolean operations of 2-manifold through vertex neighborhood classification. ACM Transactions on Graphics 1986;5(1):1–29.
[20] Masuda H. Topological operators and Boolean operations for complex-based non-manifold geometric models. Computer-Aided Design 1993;25(2):119–29.
[21] Tilove R. Set membership classification: a unified approach to geometric inter-section problems. IEEE Transactions on Computers 1980;29(10):874–83.
[22] Requicha AAG, Voelcker HB. Boolean operations in solid modeling: boundary evaluation and merging algorithms. Proceedings of the IEEE 1985;73(1):30–44.
[23] STEP ISO 10303-203, Product data representation and exchange: configuration controlled 3D designs of mechanical parts and assemblies. International Organization for Standardization, 1997.
[24] Binnard M, Cutkosky MR. A design by composition approach for layered manufacturing. ASME Transactions. Journal of Mechanical Design 2000;122(1):91–101.
[25] Armstrong MA. Basic topology. New York: Springer; 1983.
[26] ACIS geometric modeler application guide. Spatial Technology Inc.; 1999.
[27] Bhashyam S, Shin KH, Dutta D. An integrated CAD system for design of heterogeneous objects. Rapid Prototyping Journal 2000;6(2):119–35.
[28] Gorlov I. Gradient alumina–aluminide alloys (3A). http://www.tu-harburg.de/~gkig/project/fgm/abstract.htm, 1998.
[29] Kripac J. A Mechanism for persistently naming topological entities in history-based parametric solid models. Computer-Aided Design 1997;29(2):113–22.
[30] Capoyleas V, Chen X, Hoffmann CM. Generic naming in generative, constraint-based design. Computer-Aided Design 1996;28(1):17–26.
[31] Raghothama S, Shapiro V. Boundary representation deformation in parametric solid modeling. ACM Transactions on Graphics 1998;17(4):259–86.