

비다양체 형상 모델링을 위한 간결한 경계 표현 및 확장된 오일러 작업자

이상현*, 이권우**

Compact Boundary Representation and Generalized Euler Operators for Non-manifold Geometric Modeling

Sang Hun Lee*, Kunwoo Lee**

ABSTRACT

Non-manifold topological representations can provide a single unified representation for mixed dimensional models or cellular models and thus have a great potential to be applied in many application areas. Various boundary representations for non-manifold topology have been proposed in recent years. These representations are mainly interested in describing the sufficient adjacency relationships and too redundant as a result. A model stored in these representations occupies too much storage space and is hard to be manipulated. In this paper, we proposed a compact hierarchical non-manifold boundary representation that is extended from the half-edge data structure for solid models by introducing the partial topological entities to represent some non-manifold conditions around a vertex, edge or face. This representation allows to reduce the redundancy of the existing schemes while full topological adjacencies are still derived without the loss of efficiency. To verify the statement above, the storage size requirement of the representation is compared with other existing representations and present some main procedures for querying and traversing the representation. We have also implemented a set of the generalized Euler operators that satisfy the Euler-Poincare formula for non-manifold geometric models.

Key words : Geometric modeling, Non-manifold, Boundary representation, Euler operator

1. 서 론

1.1 비다양체 모델러의 필요성

기하학적 형상 모델링 시스템(geometric modeling system)은 CAD/CAM 시스템의 중심을 이루는 부분으로, 와이어프레임(wireframe)으로부터 곡면(surface) 모델러로 그리고 솔리드(solid) 모델러로 진화해 왔다. 그러나 현재 주로 사용되고 있는 형상 모델링 시스템은 설계사 도면을 작성하기 위한 2차원 제도 시스템(computer-aided drafting system) 과 자유곡면을 갖는 부품을 가공하기 위한 곡면 모델러에 불과한 실정이며, 최근 들어 솔리드 모델러를 설계에

사용하고자 하는 시도가 있으나 아직 만족할 만한 성과를 얻지 못하고 있다. 그것은 기존의 솔리드 모델이 설계가 완성된 후의 최종 결과 형상만을 표현할 수 있을 뿐, 설계 과정 중에 나타나는 불안정한 입체 형상을 표현할 수가 없으며, 설계 과정 중의 설계자의 의도를 포함하지 못하기 때문이다⁽¹⁾. 또한 엔지니어링 해석을 위해서는 솔리드 대신 보다 축약된 형태의 모델이 필요한데, 기존의 솔리드 모델러들은 이것을 지원할 수 없다. 이러한 한계점을 해결하기 위해 제안된 것이 비다양체(non-manifold) 모델러이다.

다양체(manifold) 모델이란 '어느 점에 관해서도 그 점의 근방이 평면 또는 만평면과 동상인 공간(면)'으로 표시되는 2차원 다양체(two manifold)인 곡면을 경계로 갖고 있는 부피를 가진 물체, 즉 솔리드 모델을 뜻한다. 이에 반하여 비다양체 물체란

*정회원, 고등기술연구원 생산기술연구실
**중신회원, 서울대학교 기계설계학과

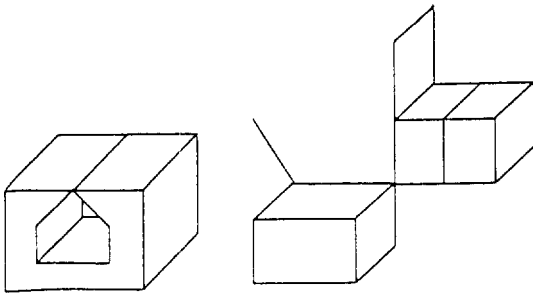


Fig. 1. Examples of non-manifold model.

Fig. 1과 같이 다양체가 아닌 것들을 뜻하며, 여기에는 와이어프레임과 솔리드의 혼합체와 같은 균일하지 않은 차원의 형상도 포함된다. 따라서 비다양체 모델러에서는 와이어프레임(wireframe) 곡면, 솔리드 모델들과 이들의 복합체 및 세포 구조의 모델이 하나의 자료 구조로서 표현이 가능하다⁽²⁾. 또한 비다양체 표현에서는 불리안 작업(Boolean operation)이 달혀 있으며, 불리안 작업의 결과를 병합체(merged set) 모델로 저장할 수가 있다. 이 병합체 모델을 사용하면 임의로 불리안 작업을 취소하거나 재 수행하기가 용이하며^(1, 3), 기존의 경계 표현(boundary representation, B-rep) 모델러에서 구현하기 복잡했던 특징 형상 기반 모델러(feature-based modeler)를 손쉽게 구현할 수도 있다⁴. 또한 비다양체 표현에서는 서로 다른 차원의 모델이 섞인 상태를 표현할 수 있으므로 개념 설계를 위한 CAD시스템을 구축하는데 사용될 수도 있다⁽⁵⁾.

1.2 비다양체 모델러의 연구 방향

비다양체 모델러와 관련된 지금까지의 연구는 크게 2가지로 나눌 수 있다. 첫번째는 비다양체 모델의 표현 방법에 관한 연구로서, 여기서는 주로 비다양

체 모델을 표현하는데 충분하면서 효과적인 위상 요소들을 창안해 내고, 이들간에 인접 관계를 저장할 수 있는 자료 구조를 개발하는 연구가 진행되어 왔다. 두번째는 비다양체 모델을 생성, 수정하는 위상 작업의 개발에 관한 연구로서, 여기서는 솔리드 모델러에서 사용되고 있는 오일러 작업자(Euler operator)⁽⁶⁾를 비다양체 영역까지 확대시키는 연구가 진행되어 왔다. 그럼으로써 오일러 작업자가 가지고 있는 장점인 형상 모델의 수학적인 정합성 유지, 자료 구조의 특정성과 복잡성으로부터 상위 레벨의 작업을 절연, 그리고 실행한 오일러 작업자에 대한 역작업자를 사용한 취소 명령 구현의 용이와 같은 장점을 비다양체 모델러에서도 취할 수가 있게 되었다. 그러면 지금까지의 이들에 관한 연구에 대하여 살펴 보도록 하겠다.

1.2.1 경계 표현 방법

비다양체 모델의 표현에 대한 최초의 팔목할 만한 연구는 Weiler⁽²⁾에 의해서 수행되었다. Weiler는 물체의 모서리에 인접한 면들을 순서적으로 저장한 Radial Edge Structure라는 경계 표현 방식의 자료 구조를 제안하였다. 일반적으로 인접한 위상 요소들을 일정한 순서에 따라 저장하는 정렬된 형태의 비다양체 표현 방식에서는 Fig. 2에 나타난 바와 같이 루프 순환(loop cycle), 방사형 순환(radial cycle) 그리고 디스크 순환(disk cycle)의 3가지 순환 정보를 가지고 있어 필요시 이를 효과적으로 사용하는데^(7, 8), Weiler의 경우는 이 가운데 루프와 방사형 순환을 저장하는 최초의 비다양체 자료 구조이다.

Choi⁽⁷⁾는 zone 과 disk라는 위상 요소를 추가로 도입하여, Weiler의 자료 구조에서 불완전하게 처리되었던 꼭지점에서의 이웃 관계를 명확히 나타내는 Vertex-Based B-rep을 제안하였다. 이 자료 구조에서는 루프 순환과 방사형 순환뿐만 아니라 디스크 순

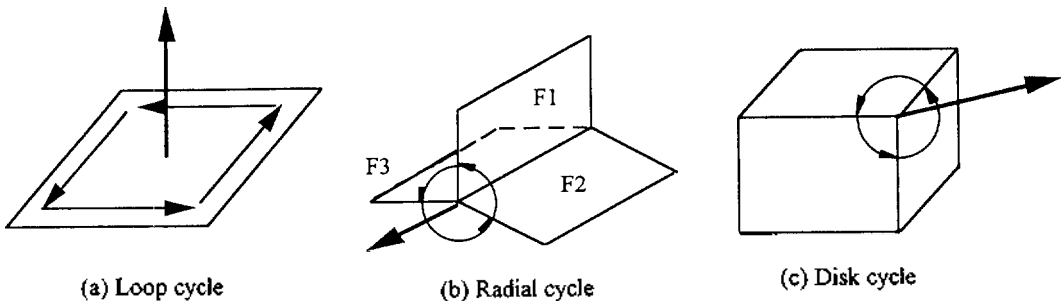


Fig. 2. Three cycles in non-manifold B-rep.

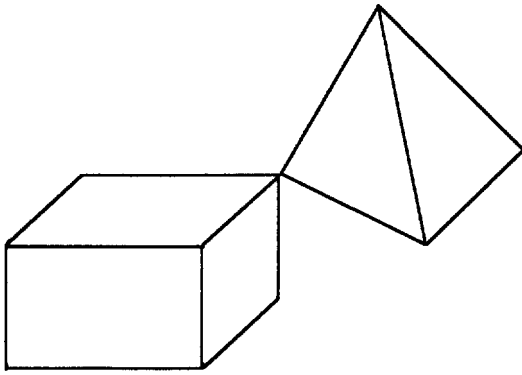


Fig. 3. Example of r-set model.

환을 명확히 저장하고 있다. Yamaguchi⁽⁸⁾는 꼭지점, 모서리, 면 그리고 영역(region)의 기본적인 위상 요소들의 인접 관계를 나타내기 위해 6가지의 coupling entity들을 새로 도입하여, 이웃 관계에 바탕을 둔 다양체 표현을 제시하였으며, 여기에서도 위에서 언급한 3가지 사이클을 내포하고 있다. 그 밖에 Rossignac⁽⁹⁾은 서로 해체되어 있는 셀들의 유한 개의 집합으로 구성되어 있는 Selective Geometric Complex (SGC)를 제안하였다. 이 셀들은 n차원 다양체들의 열린 부분 집합(open subset)으로 꼭지점(0차원), 모서리(1차원), 면(2차원), 그리고 영역(3차원)의 개념을 일반화시킨 것으로 이들 셀들간의 연결은 아주 간단한 인접 그래프를 사용하고 있다. 따라서 이 SGC는 모델링 영역을 일반화시키고 저장 장소를 줄인 반면, 셀들간의 경계 관계와 위상 정보를 얻어내는 데는 비효율적이다.

1.2.2 위상 작업

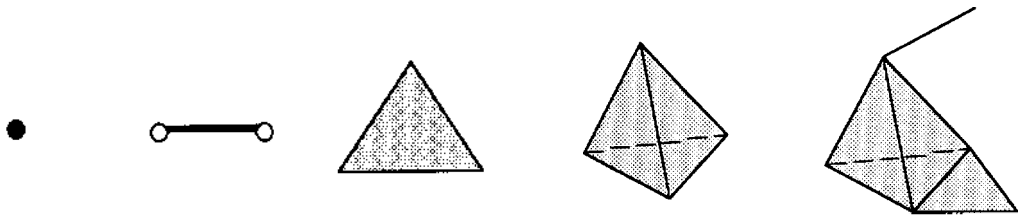
일반적인 오일러-포앙카레 공식(Euler-Poincare formula) (이하 오일러 공식)으로부터 유도된 위상 요소간의 관계식은 그 표현 범위에 따라 크게 두 가

지로 나눌 수 있다. 그 하나는 regular set(r-set) 모델 까지 확장된 오일러 공식으로 이 분야의 연구는 Luo⁽¹⁰⁾와 Heisserman⁽¹¹⁾에 의해 행해졌으며, 다른 하나는 3차원 유클리드 포복체(Euclidean cell complex) 까지 적용시킬 수 있는 오일러 공식으로 Masuda⁽¹²⁾와 Yamaguchi⁽⁸⁾등에 의해서 제안되었다.

여기서 r-set 모델은 경계지워지고(bounded), 닫힌(closed), 정규의(regular) 3차원 유클리드 공간(E³)의 부분 집합(subset)들이 모여서 이루어진 모델로서⁽⁶⁾, 여기서 정규의(regular) 의미는 어떤 point set이 자신의 내부점들의 closure인 경우를 뜻한다. 이러한 r-set 모델은 Fig. 3에서 나타난 바와 같이 솔리드들의 집합으로 나타나게 되며, 여기에서는 와이어프레임이나 박판 면(laminar face)은 존재하지 않는다.

또한 유클리드 포복체(Euclidean cell complex)는 3차원 유클리드 공간(E³)에 존재하는 0차원부터 3차원까지의 유클리드 포체(Euclidean cell)들의 집합으로 이루어진 물체를 뜻한다. 여기서 0차원부터 3차원까지의 각 유클리드 포체는 Fig. 4에 나타난 것과 같으며, 이들 각각의 포체는 기하학적 모델에서 사용되는 위상 요소인 꼭지점(vertex), 모서리(edge), 면(face), 그리고 영역(region)과 일치한다. 즉, 0차원 포체(0-cell)는 꼭지점, 1차원 포체(1-cell)는 모서리, 2차원 포체(2-cell)는 면, 그리고 3차원 포체(3-cell)는 영역에 각각 해당된다.

Luo⁽¹⁰⁾는 평면(plane) 모델을 r-set 모델에 맞는 다중 평면(multiplane) 모델로 확장시켰으며, 세 가지 변수를 추가로 도입하여 r-set 모델에 적용될 수 있는 확장된 오일러 공식을 제안하였다. 따라서 이를 바탕으로 한 오일러 작업자는 다양체 모델에 대한 기존의 5가지 오일러 작업자에 r-set 모델을 위한 3가지 작업자를 추가시킨 것이 되었다. 또한 Heisserman⁽¹¹⁾은 Luo와 마찬가지로 r-set 모델에까지 적용



(a) 0-Cell (Vertex) (b) 1-Cell (Edge) (c) 2-Cell (Face) (d) 3-Cell (Region) (e) Cell complex

Fig. 4. Euclidean cells and a cell complex.

시킬 수 있는 오일러 공식을 제안하였다. 그러나 이들의 접근 방식은 r-set 영역에 제한된 것으로 와이어 프레임 모서리(이하 와이어 모서리)나 박판 면이 혼합된 일반적인 비다양체 모델에까지는 적용시킬 수 없다.

한편 Masuda⁽¹⁾는 3차원 유클리드 공간 내의 포복체에 적용시킬 수 있는 확장된 오일러 공식을 제시하였으며, 이식을 바탕으로 하여 9쌍의 최소한의 비다양체 오일러 작업자들을 제안하였다. 그러나 그가 제시한 오일러 공식에서는 빈 영역과 채워진 영역에 해당하는 물체의 volume과 cavity가 개별적인 변수로 사용됨으로써, 자료 구조의 대칭성이 깨지고 알고리즘이 대단히 복잡해지는 단점을 가지고 있으며, 또한 모델링 작업을 효과적으로 구현하기 위해서는 그가 제시한 최소한의 오일러 작업자 외에 추가적인 오일러 작업자가 필요하다.

Yamaguchi⁽⁶⁾는 Masuda와 달리 volume과 cavity를 모두 region으로 나타내며, 이를 바탕으로 비다양체 모델에 대한 오일러 공식을 제안하였다. 또한 그들은 기본적인 위상 요소들인 꼭지점, 모서리, 면 그리고 영역의 이웃 관계를 나타내는 coupling entity들을 위상 요소로서 추가로 도입하였으며, 이 coupling entity들간의 구속 조건을 만족하는 하위 레벨의 neighborhood operation들을 제시하였으며, 이 하위 작업들을 사용하여 오일러 작업을 구현하였다. 그러나 그들은 이식을 근거로 한 모델링에 필요한 최소한의 오일러 작업자를 제시하지 않고, 단지 몇 가지 가능한 오일러 작업자들의 예만을 나열하는 데 그쳤다.

1.3 본 논문의 목적

앞에서 살펴본 바와 같이 지금까지의 비다양체 모델에 대한 자료 구조들은 주로 각 위상 요소들간의 인접 관계들을 충분히 저장하는 데 초점이 맞추어져 있기 때문에 복잡하며 위상 정보들이 지나치게 중복되어 있다. 이는 위의 Weiler⁽²⁾, Choi⁽⁷⁾ 그리고 Yamaguchi⁽⁶⁾의 표현 방식에서 표현하고자 하는 초점이 3차원 위상 요소인 각 영역(region)의 경계에 있기 때문에 여러 개의 영역으로 구성된 모델인 경우 각각의 영역이 다양체 솔리드 모델과 동등하게 완전한 경계 정보를 저장하고 있으며, 따라서 인접한 영역들과 서로 중복되는 경계 정보를 저장하게 되기 때문이다. 이러한 자료 구조의 중복성은 기억 저장 장소의 낭비를 가져오며, 또한 복잡한 자료 구조와 이를 조작하기 위한 프로그램의 개발의 난점을 가져오

는 요인이 되어 왔다.

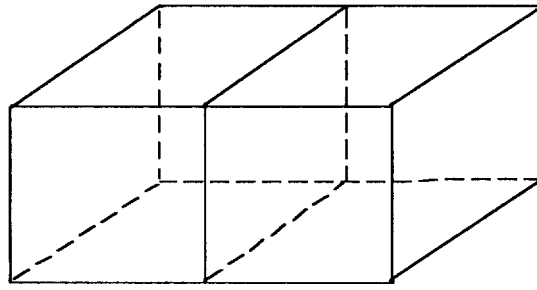
본 논문은 이와 같은 앞의 표현들이 가진 지나친 자료의 중복성을 극복하기 위해 새로운 형태의 비다양체 모델을 위한 간결한 경계 표현 방식을 제안하였다. 이 방식은 다양체 모델을 위한 half-edge data structure⁽⁶⁾ 표현 방법으로부터 출발하여 모서리에 n개의 면들이 인접해 있는 경우, 꼭지점에 n개의 꼭지점들이 인접해 있는 경우, 면의 양쪽에 두 영역이 인접해 있는 경우와 와이어 모서리가 존재하는 경우 등 비다양체적인 상태를 수용할 수 있도록 부분 면(partial face), 부분 모서리(partial edge), 부분 꼭지점(partial vertex)과 같은 새로운 위상 요소를 도입하고 자료 구조를 확장했다. 따라서 기존의 경계 표현 방식들이 각각의 영역에 대한 완전한 경계 및 이웃 정보를 저장하는데 반해서, 이 새로운 경계 표현법에서는 면들에 대한 완전한 경계 및 이웃 정보를 저장하고 있으며 영역에 대한 경계 및 이웃 정보는 이로부터 추출해 내도록 되어 있다. 이에 대한 자세한 설명은 뒤의 2.1.1장에서 하도록 하겠다.

또한 본 논문에서는 비다양체 모델의 영역까지 확장된 오일러 공식을 바탕으로 모델러 개발에 필요한 충분하면서도 최소한의 오일러 작업자를 제안하고 구현하였다. 모델링 과정중에도 항상 모델이 항상 수학적 정합성을 유지하고, 상위 레벨의 프로그램을 자료 구조의 특정성과 복잡성으로부터 절연시키며, 각 오일러 작업자에 역작업자를 제공함으로써 취소 명령을 쉽게 구현할 수 있도록 하였다.

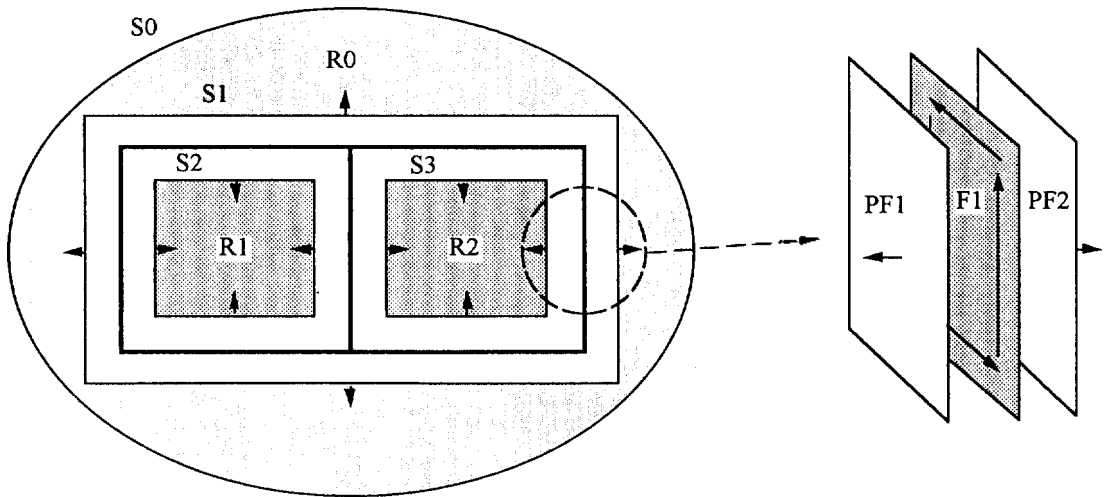
2. 비다양체 모델의 경계 표현

2.1 위상 요소

본 연구에서 다루는 대상 물체는 Fig. 1에 나타난 것과 같은 비다양체 물체로서, 앞의 1.2.2장에서 설명한 3차원 유클리드 포복체(Euclidean cell complex)이다. 본 비다양체 경계 표현 방식에서 이들을 나타내기 위하여 먼저 0차원에서 3차원까지의 유클리드 포복체에 해당되는 위상 요소로서 꼭지점(vertex), 모서리(edge), 면(face), 그리고 영역(region)을 도입하였다. 즉, Fig. 4에서와 같이 0차원 포복체(0-cell)는 꼭지점, 1차원 포복체(1-cell)는 모서리, 2차원 포복체(2-cell)는 면, 그리고 3차원 포복체(3-cell)는 영역에 각각 해당된다. 또한 면과 영역의 경계를 나타내기 위한 위상 요소로서 각각 루프(loop)와 셸(shell)을 도입하였다. 이상의 위상 요소들은 기하학적 형상 모델을 표현하기 위한 기본적인 위상 요소들로서 앞의 Weiler⁽²⁾,



(a) A cube with an inner partition



R0, R1, R2 : Region
S0, S1, S2, S3 : Shell

PF1, PF2 : Partial Face
F1 : Face

(b) Region and shell

(c) Partial face

Fig. 5. Example of regions shells and partial faces.

Choi⁽⁷⁾, Yamaguchi⁽⁸⁾등의 논문에서 사용한 것과 동일하게 정의된다.

그밖에 본 논문에서는 이들 기본적인 위상 요소들 간의 인접 관계들을 나타내기 위하여 부분 면(partial face), 부분 모서리(partial edge) 그리고 부분 꼭지점(partial vertex)의 3가지 독자적인 위상 요소들을 도입하였다. 이들은 각각 면과 영역, 모서리와 면, 그리고 꼭지점과 모서리간의 인접 관계를 나타내는 데 사용되며, 이들을 통칭하여 부분 위상 요소(partial topological entity)라 부른다. 여기서 부분 면은 면의 양쪽에 두 영역이 인접해 있는 경우, 부분 모서리는 모

서리에 n개의 면들이 인접해 있는 경우, 끝으로 부분 꼭지점은 꼭지점에 n개의 꼭면들이 인접해 있는 경우의 비다양체적인 상황을 표현 가능하도록 해주는 역할을 함께하고 있다. 그러던 이들 각 부분 위상 요소들에 대하여 상세히 설명하도록 하겠다.

2.1.1 부분 면

비다양체 모델은 일반적으로 무한대의 크기를 갖는 하나의 영역과 0혹은 그 이상 개수의 닫힌 영역들로 이루어져 있다. 예를 들어 Fig. 5(a)와 같이 하나의 칸막이를 갖는 상자형 물체가 있을 때, Fig. 5(b)에 나타난 바와 같이 상자 외부의 무한대 영역을

나타내는 영역인 R0와 상자의 내부를 나타내는 두 개의 영역 R1, R2의 세 영역이 존재한다. 이들 각각의 영역들은 방향성을 갖는 경계 위상 요소인 셸에 의해 경계지어지는데, 모든 영역은 하나의 바깥쪽 셸(peripheral shell)을 갖으며, 자신의 내부에 다른 영역이 있는 경우에는 안쪽 셸(void shell)을 추가로 갖게 된다. Fig. 5(b)에서 S0는 무한대 영역인 R0의 가상적인 바깥쪽 셸이며, S1은 R0의 안쪽 셸이고, S2와 S3는 각각 R1과 R2의 바깥쪽 셸이다. 비다양체 모델인 경우 면은 항상 2개의 영역과 인접해 있으며, 면의 양쪽 측면은 각 영역의 경계인 셸의 한 부분을 이루어야 한다. 이를 위해 하나의 면을 둘로 쪼개서 만든 것이 부분 면으로서, 이들이 모여서 하나의 셸을 이룬다.

부분 면은 면에 대한 포인터와 방향 표시로써 면의 양쪽 측면 가운데 하나를 나타내며 방향성을 가지고 있다. 이때 부분 면의 방향은 Fig. 5(b)에 나타난 바와 같이 영역의 안쪽 방향을 향하도록 되어 있다. 여기서 부분 면의 방향이란 부분 면의 곡면의 법선 방향을 가리키는 것으로서 이는 부분 면에서 저장하고 있는 방향 표시 플래그가 면과 같은 쪽으로 지정되어 있으면 자신이 속한 면의 곡면의 법선 방향과 일치하고, 그렇지 않으면 면의 곡면의 법선 방향과 반대가 된다. 예외적인 경우로서 면이 아닌 공

간상에 고립된 꼭지점이나 와이어 모서리가 존재할 때는 그 꼭지점 또는 모서리에 대한 포인터만 가지는 방향성이 없는 부분 면이 각각 할당되며, 그럼으로써 와이어프레임 모델도 하나의 자료 구조 내에서 통일성 있게 표현될 수 있다.

이 부분 면은 개념적으로는 Weiler⁽²⁾와 Choi⁽⁷⁾의 face-use, Yamaguchi⁽⁸⁾의 side와 동일하지만 부분 면의 경계(boundary)를 표현하는 데 있어서는 앞의 표현 방식들과는 차이가 있다. 즉, Weiler의 경우는 Fig. 6에 나타난 바와 같이 면이 자신의 경계에 대한 자료를 가지고 있지 않고, 그 면에 대한 두 face-use가 자신의 경계를 edge-use의 집합인 각각의 loop-use를 통해서 완전하게 저장하고 있다. 따라서 방향만 다른 두개의 면 경계에 대한 자료가 중복되어 저장되게 된다. Choi의 wall과 Yamaguchi의 side도 Weiler의 face-use와 동일한 위상 요소이다.

이에 반해 본 논문의 자료 구조에서는 부분 면이 아닌 면이 자신의 경계에 대한 정보를 가지고 있으며, 부분 면의 경계에 대한 정보가 필요할 때는 소속된 면의 경계 정보로부터 추출해 내는 방식을 취하고 있다. 또한 face-use는 인접한 face-use에 대한 정보를 가지고 있는데 반하여, 부분 면은 자신과 인접한 부분 면과의 관계를 저장하고 있지 않고, 소속면의 인접 정보로부터 유도해 낸다. 따라서 본 논문의

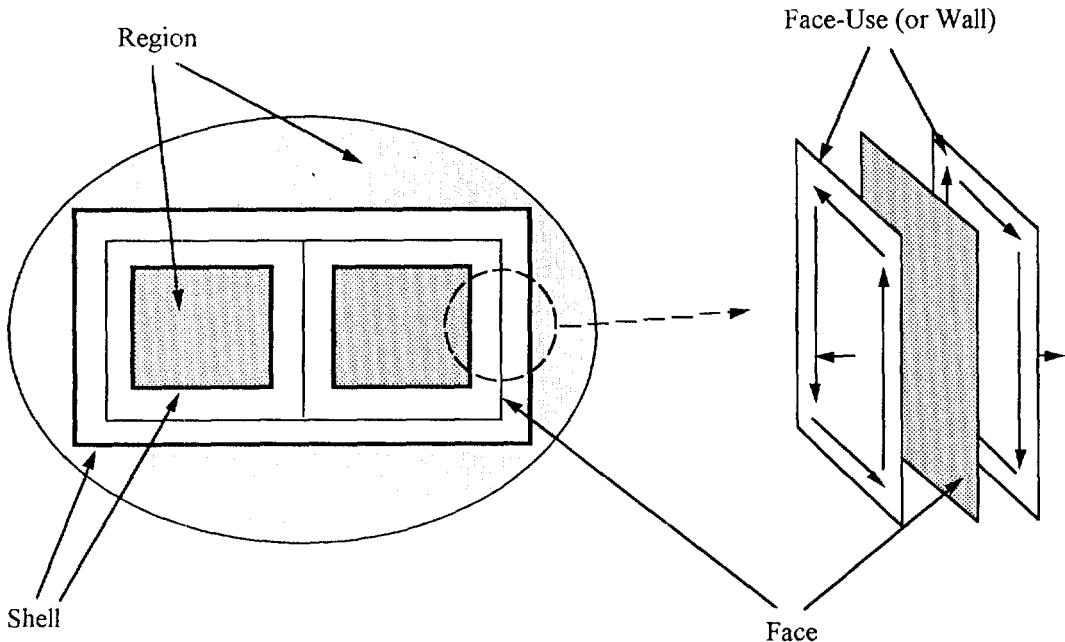


Fig. 6. Example of face-uses in Weiler's B-rep.

방식은 부분 면이 자신과 인접한 부분 면을 찾고자 할 때는 Weiler의 방식보다 다소 시간이 더 걸리는 반면, 면의 경계에 대한 중복되는 자료를 없애는데 효과적이다.

2.1.2 부분 모서리

다양체 모델에서는 하나의 모서리는 항상 2개의 면과 인접하고 있는데 반하여, 비다양체 모델일 경우에는 임의의 개수의 면과 인접하게 된다. 이때 하나의 모서리를 인접한 면의 개수만큼 쪼개어 모서리와 면과의 인접 관계를 표현한 것이 부분 모서리이다. 따라서 Fig. 7에서 나타난 바와 같이 부분 모서리는 모서리에 대해서 인접한 면들의 개수만큼 나타난다. 즉, 모서리 E1은 인접한 세면 F1, F2, F3에 대해서 각각 PE1, PE2, PE3의 세 개의 부분 모서리를 할당해 준다. 이때 모서리의 둘레에 있는 부분 모서리들은 모서리의 방향에 대해서 오른손 법칙에 따른 순서로 정렬되어 있다. 즉, 오른손 엄지 손가락을 모서리의 방향으로 두고 주먹을 쥐었을 때, 나머지 네 손가락의 방향을 따라 부분 모서리들이 차례로 정렬되어 있다. 이 정렬은 앞에서 말한 모서리에서의 방

사형 순환을 나타내며, 이 정보는 뒤에 새로운 셀을 구성하는 부분 면들을 찾아내는데 유용하게 쓰인다. Fig. 7에서 ⊙표시는 논문 종이면의 바깥으로 나오는 방향을 뜻하며, 본 논문 그림에서 나오는 ⊗표시는 종이면의 안쪽으로 들어가는 방향을 뜻한다.

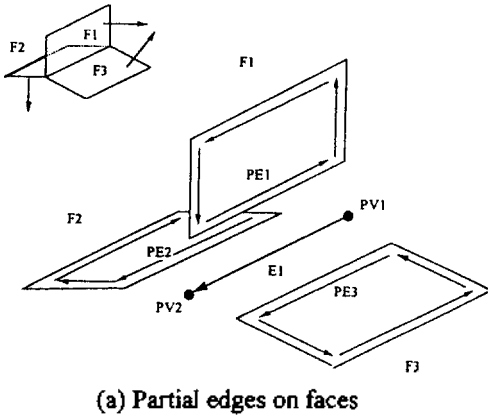
부분 모서리는 보통 자신이 속한 모서리와 방향으로 표현되며, 인접한 면의 경계인 루프를 구성하는 요소가 된다. 이때 부분 모서리들의 방향은 루프의 방향과 일치하며, 루프내에 차례로 정렬되어 있다. 이러한 정렬은 앞에서 말한 루프 사이클을 의미한다. 이때 예외적인 경우로서 고립된 꼭지점이 하나의 루프를 형성하는 경우가 있는데, 이 때에는 자료 구조의 통일성을 위해서, 꼭지점으로 향하는 포인터만을 가지는 방향성 없는 부분 모서리가 할당된다.

이 부분 모서리는 상용 형상 모델링 시스템인 ACIS에서 사용하고 있는 co-edge와 유사한 위상 요소이다⁽¹²⁾. 그러나 ACIS에서는 와이어 모서리를 포함한 모든 종류의 모서리에 co-edge를 붙인 반면, 우리의 자료 구조에서는 면의 경계를 이루는 모서리들에 대해서만 부분 모서리를 붙이며, 따라서 모서리에 인접한 면의 개수만큼의 부분 모서리가 생성된다. 또한 Weiler가 사용한 edge-use와 비교해 보면⁽²⁾, Weiler의 edge-use는 face-use를 경계짓는 loop-use의 구성 요소인데 반하여, 이 부분 모서리는 face를 경계짓는 loop의 구성 요소이다. 따라서 한 모서리에 대해서 부분 모서리는 인접한 면의 개수만큼 존재하는 반면, Weiler의 edge-use는 인접한 면의 개수의 두 배만큼 존재한다.

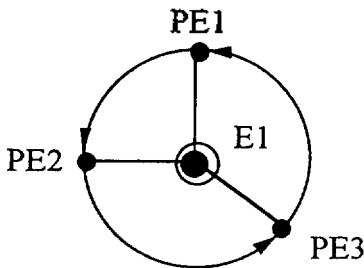
2.1.3 부분 꼭지점

부분 꼭지점은 꼭지점에 복수 개의 다양체(two manifold) 곡면이 인접해 있는 비다양체적인 상황을 지원하기 위해서 도입되었다. 여기서 다양체 곡면은 하나의 면 혹은 서로 연결된 하나 이상의 면들의 집합으로 이루어지며, 예외적인 경우로 와이어 모서리는 하나의 곡면이 퇴화한 것으로 간주한다. Fig. 8에 나타난 바와 같이, 부분 꼭지점은 꼭지점에 인접한 곡면에 대해서 하나씩 만들어져 있다. 즉, 꼭지점 V1은 두 곡면에 대해 PV1과 PV3를, 그리고 와이어 모서리 E2에 대해 PV2를 할당해 주고 있다. 다양체 모델인 경우에는 하나의 꼭지점에 하나의 곡면이 인접해 있으므로 항상 하나의 부분 꼭지점만을 가지지만, 비다양체 상황에서는 하나 이상의 부분 꼭지점들을 가질 수 있다.

부분 꼭지점은 자신이 속한 꼭지점과 하나의 모서리에 대한 포인터를 저장하고 있다. 여기의 모서리



(a) Partial edges on faces



(b) Cyclic ordered partial edges around an edge

Fig. 7. Example of partial edges.

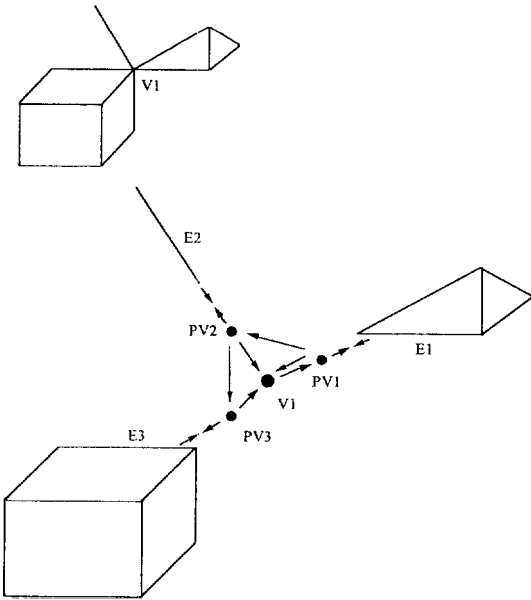


Fig. 8. Example of partial vertices around a vertex.

는 인접한 곡면 상에 있는 한 모서리로서, 꼭지점을 자신의 한 끝점으로 갖고 있는 모서리이다. 예외적인 경우로서 고립된 하나의 꼭지점이 하나의 루프를 형성하는 경우가 있다. 이 경우에도 인접한 곡면은 하나이므로 하나의 부분 꼭지점이 그 꼭지점에 할당되고, 루프도 또한 자신의 구성 요소인 부분 모서리를 하나 갖게 된다. 그러나 이 경우에는 부분 꼭지점과 부분 모서리를 연결해 주는 모서리가 없기 때문에 부분 꼭지점은 직접 부분 모서리를, 부분 모서리는 직접 부분 꼭지점을 가리키도록 하였다.

이 부분 꼭지점은 본 자료 구조의 독특한 위상 요소로서, Weiler의 vertex-use등과는 상당히 다르다⁽²⁾. Vertex-use는 edge-use의 끝점을 나타내며, 한 꼭지점에 대하여 인접한 face-use와 와이어 모서리의 개수만큼 존재한다. 이에 반해 부분 꼭지점은 모서리의 끝점을 나타내며, 인접한 곡면과 와이어 모서리의 개수만큼 존재한다.

2.2 자료 구조

비다양체 모델을 표현하기 위한 자료 구조를 설계할 때에는 다음의 세 가지 퇴화적인 경우(degenerate case)를 수용할 수 있도록 해야 한다; (1)공간상에 하나의 고립된 꼭지점만이 놓여 있어 이 꼭지점만으로 셸을 형성하는 외꼭지점 셸(single vertex shell)인 경우, (2)모서리가 어떤 면과도 인접해 있지 않은 와이어 모서리(wire edge)인 경우, (3)면 위의 하나의 고

립된 꼭지점이 놓여 있어 이 꼭지점만으로 내부 루프를 형성하는 외꼭지점 루프(single vertex loop)인 경우가 이에 해당된다. Fig. 9은 위의 세 가지 예외적인 경우를 포함한 각 위상 요소들간의 관계를 계층 구조적인 자료 구조를 사용하여 표현한 것이며, C++ 프로그래밍 언어를 사용한 자료 구조의 구현은 부록 A에 소개되어 있다.

Fig. 9의 자료 구조에서 (a)부분은 일반적인 경우를 나타내는 자료 구조이고, (b)(c)(d)부분은 위의 세 가지 퇴화적인 경우를 통일된 자료 구조로 표현하기 위하여 예외적으로 처리되는 것을 나타내고 있다. 즉, (b)부분은 외꼭지점 루프의 경우로서, 루프를 구성하는 하나의 부분 모서리는 자신이 속하는 모서리가 하나의 꼭지점으로 퇴화함으로써 모서리로 향하는 포인터 대신 퇴화된 꼭지점으로 향하는 포인터만을 가지는 방향성 없는 부분 모서리이며, 꼭지점의 부분 꼭지점도 인접한 모서리가 없으므로 상위 위상 요소로서 모서리에 대한 포인터 대신에 루프의 부분 모서리의 포인터를 저장한다. 또한 (c)와 (d)부분은 외꼭지점 셸과 와이어 모서리의 경우로서, 이때 셸을 구성하는 부분 면은 본래 자신이 속한 면이 꼭지점 혹은 와이어 모서리로 퇴화함으로써 자신이 속한 면 대신에 퇴화된 꼭지점 또는 모서리로 향하는 포인터만 가지는 방향성이 없는 부분 면이 되며, 각각의 꼭지점과 모서리도 상위 위상 요소로서 부분 면을 가리키게 된다.

2.3 자료 탐색

2.3.1 인접 관계의 추출

Weiler⁽²⁾ 및 Choi⁽⁷⁾의 자료 구조와 본 자료 구조에서 공통으로 사용되고 있는 기본적인 위상 요소들은 꼭지점, 모서리, 루프, 면, 셸, 영역의 6가지이므로 이들간의 인접 관계는 Table 1에 나타난 바와 같이 총 36가지가 가능하게 된다. 그런데, 일반적으로 모델러에 사용되는 자료 구조들은 중복되는 정보를 최소화시키기 위해, 모든 인접 관계에 대한 정보를 직접 저장하지 않고, 일부 인접 관계들만을 저장하고 있다. 필요한 경우 나머지 인접 관계들을 유도해 내도록 하고 있다. 본 자료 구조에서는 Table 1에서 실선 및 점선의 사각형으로 테두리 쳐진 인접 관계들만을 저장하고 있으며, Weiler의 자료 구조에서도 이와 동일한 인접 관계를 자료 구조에 저장시키고 있다. 따라서 본 자료 구조로부터 기본적인 위상 요소들간의 모든 인접 관계에 대한 정보를 추출해 낼 수 있으며, 이들 정보를 추출해 내는데 걸리는 시간은 Weil-

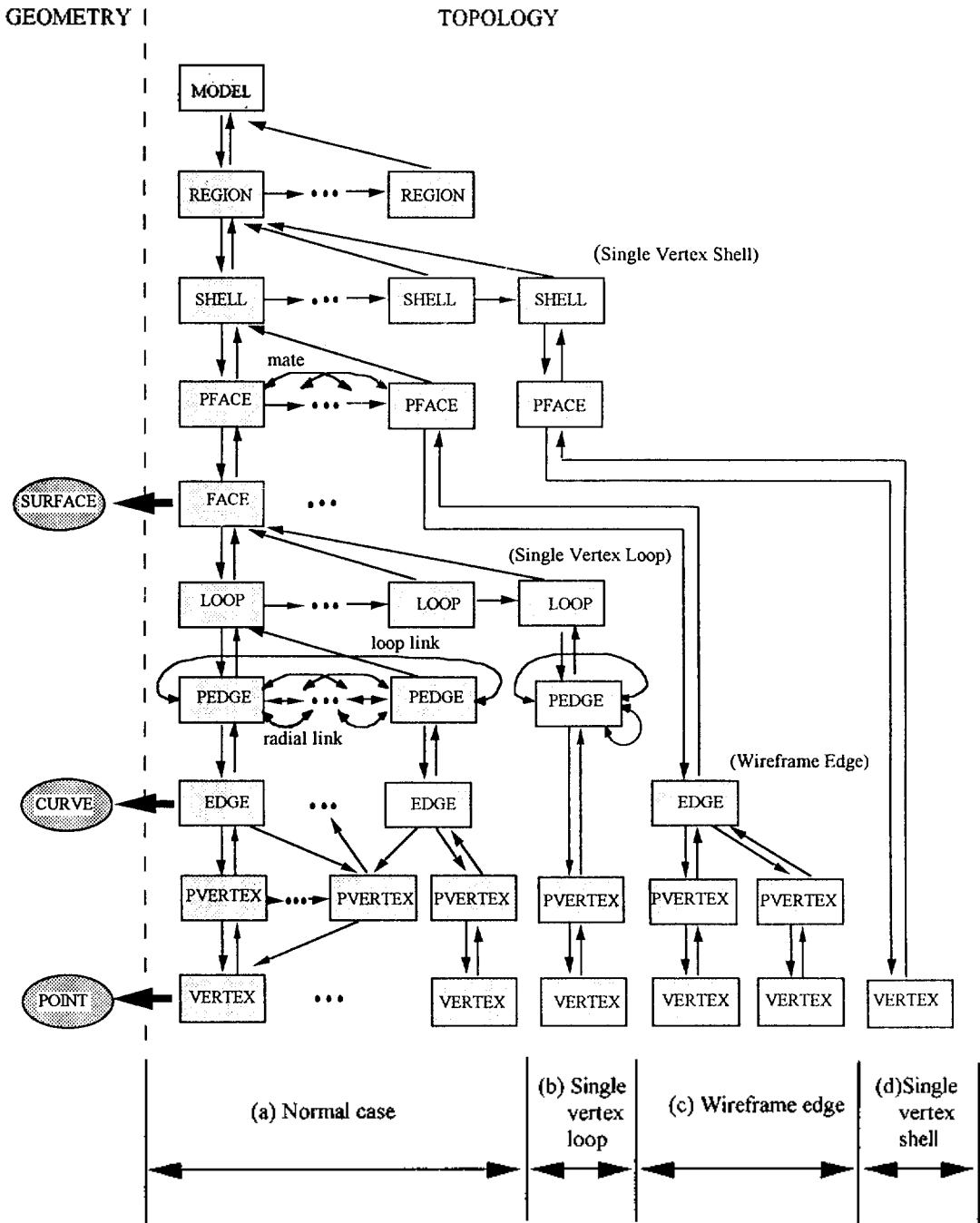


Fig. 9. Data structure of non-manifold B-rep.

er의 경우와 같은 정도의 시간을 필요로 한다는 것을 직관적으로 판단할 수 있다.



2.3.2 인접한 부분 면 찾기

비다양체 모델링 환경에서는 새로운 면을 추가시

켰을 때 새로운 영역이 생성되는지 확인하는 작업이 필요하다. 특히 오일러 작업자를 사용하여 모델링을 해 나갈 때에는 어떠한 작업자를 선택해야 할 것인가를 작업의 실행에 앞서 결정해야 한다. 이 경우 면

Table 1. Adjacency relationship matrix for our B-rep

reference element type	adjacent group element type					
	vertices	edges	loops	faces	shells	regions
vertex	V{V}	V{E}	V{L}	V{F}	V{S}	V{R}
edge	E{V}	E{<E>}	E{<L>}	E{<F>}	E{<S>}	E{<R>}
loop	L{V}	L{<E>}	L{<<L>>} ²	L{F}	L{S} ²	L{R}
face	F{<<V>>} ²	F{<<E>>} ²	F{L}	F{<<F>>} ²	F{S}	F{R}
shell	S{V}	S{E}	S{L}	S{F}	S{S}	S{R}
region	R{V}	R{E}	R{L}	R{F}	R{S}	R{R}

where { } linear group
 [] unordered group
 < > circular ordered group
 adjacency relationship stored in our B-rep for normal case
 adjacency relationship stored in our B-rep for three exceptional cases; single vertex shell, wire-frame edge, and single vertex loop

저 새로운 면이 만들어질 곳에 미리 가상의 면이 존재한다고 가정하고, 그 가상 면의 한쪽 부분 면으로부터 출발하여 인접한 부분 면들을 방문하고 다시 그 부분 면들로부터 다음 인접한 부분 면들을 차례로 방문한다. 만일 출발한 부분 면의 맞은편 짝이 되는 부분 면을 방문하지 않고 여행이 끝났다면, 새로운 면을 추가시켰을 때 새로운 영역이 생기게 된다. 따라서 이 경우에는 면과 영역을 동시에 생성시키는 오일러 작업자인 MFR을 적용하고, 그렇지 않으면 면만 생성시키는 MFKC를 적용하게 된다. 한편, MFR의 프로그램 내부에서도 기존의 영역을 둘로 나누기 위해서는 새로 만들어진 면을 경계로 바깥쪽 셀을 둘로 나누어야 하며, 이를 위해서는 새로 생긴 면의 부분 면에서 출발하여 인접한 부분 면을 방문하는 작업이 필요하게 된다. 여기서 MFR, MFKC 및 MFR을 포함한 오일러 작업자들에 대한 자세한 설명은 뒤의 3.2장에 나타나 있다.

Weiler⁽²⁾의 Radial Edge Structure인 경우에는 edge-use들간의 radial pointer를 갖고 있기 때문에 단지 일련의 포인터만 따라가면 인접한 face-use로 건너갈 수 있다. Choi⁽⁷⁾의 경우에는 Fig. 10에서와 같이 한 모서리에 대해 두개의 edge orientation이 있고, 각각의 edge orientation에 대해서 그 edge orientation과 같은 방향을 갖는 edge-use들의 사이클이 저장되어 있다. 따라서 다음의 인접 face-use를 찾는 방법은 먼저 현재의 edge-use와 같은 edge에서 파생된 edge-use 가운데에서 현재의 edge-use가 속한 face-use와 서로 짝이 되는 face-use상에 놓여 있는 edge-use, 즉 mate edge-use를 찾은 다음, 그 mate edge-use가 속한

edge orientation 사이클상의 다음 번 edge-use를 찾는다. 그 다음 그것이 속한 face-use를 찾으면 이것이 바로 원래의 edge-use가 속한 face-use의 이웃한 face-use가 된다. 그러나 본 자료 구조에서는 그러한 정보는 가지고 있지 않기 때문에 다음과 같은 몇 단계의 과정을 거쳐 다음의 인접한 부분 면으로 건너가게 된다.

본 논문에서 한 부분 면에 대한 인접한 부분 면을 찾는 알고리즘은 Choi의 자료 구조에서 인접한 face-use를 찾아가는 과정을 모사시킨 것으로, Fig. 10의 모델에 대한 적용 예가 Fig. 11에 그림으로 나타나 있다. 여기서 가상 edge-use는 부분 모서리에 대한 포인터와 부분 모서리의 방향을 기준으로 한 방향 플래그(orientation flag)을 저장하고 있다. 이 알고리즘에 대한 C++ 언어 프로그램은 부록 B에 수록되어 있다.

[인접 부분 면 찾기 알고리즘]

(1단계) 현재의 부분 면(예를 들어 pf1)의 가장자리에 가상 edge-use(eu1)를 가정한다. 이 eu1의 방향은 현 부분 면의 방향이 면의 방향과 일치하면 부분 모서리(pe1)과 같은 방향, 그렇지 않으면 반대 방향이 된다.

(2단계) eu1의 방향과 반대인 가상의 mate edge-use(eu2)를 만든다(Fig. 11(b)의 ①과정).

(3단계) Choi의 edge-orientation cycle을 따라갈 때, eu2의 다음에 이웃한 가상 edge-use(eu3)를 다음과 같이 결정한다(Fig. 11(b)의 ②과정). eu3의 방향 껏 말은 eu2와 같은 방향을 가리키도록 정하고, eu3가 갖고있는 부분 모서리에 대한 포인터는 eu2의 방향

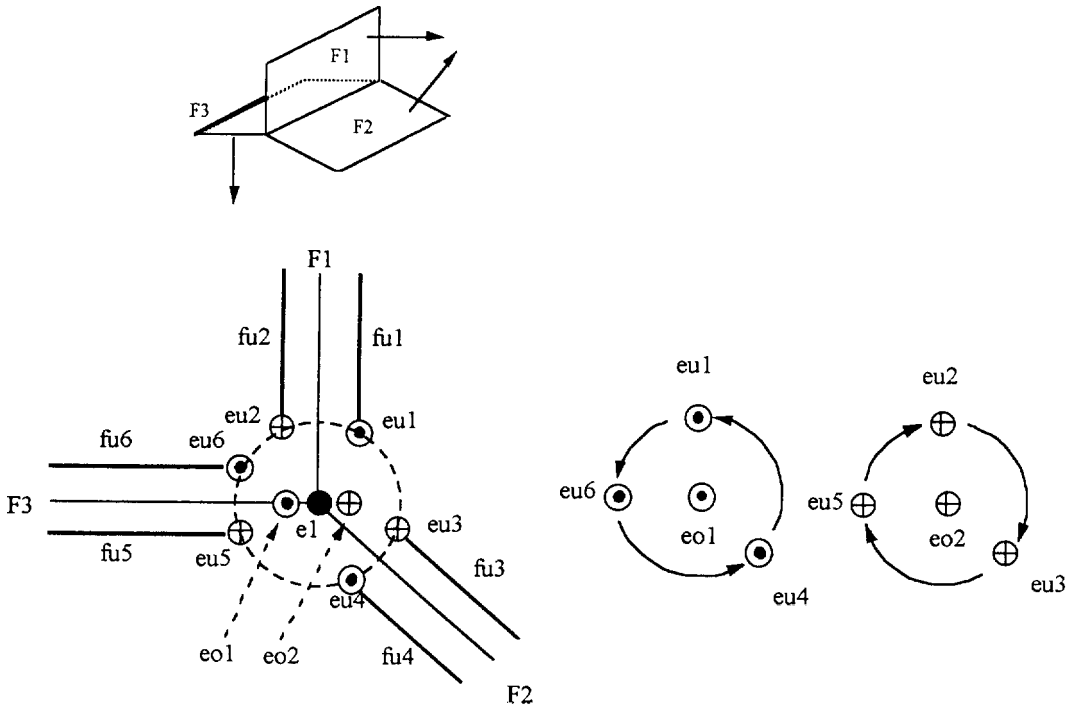


Fig. 10. Search algorithm for the next adjacent face-use in Vertex-based B-rep.

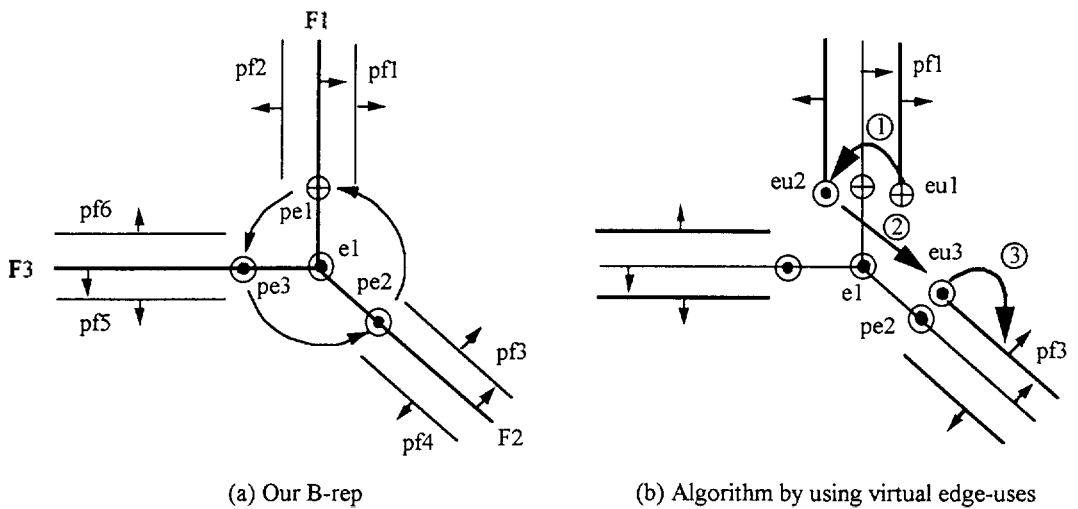


Fig. 11. Search algorithm for the next adjacent partial face in our B-rep by using virtual edge-uses.

이 모서리 e1의 방향과 일치하면 pe1의 radial_prev에 있는 부분 모서리를, 그렇지 않으면 radial_next에 있는 부분 모서리를 가리키도록 한다.

(4단계) eu3로부터 pf1의 다음에 방문할 부분 면(pf3)을 구한다(Fig. 11(b)의 ③과정). eu3의 부분 모서리인 pe2의 방향이 eu3와 같으면 면과 같은 방향

의 부분 면, 다르면 면과 반대 방향의 부분 면이 다음에 방문해야 할 부분 면이다.

Weiler의 자료 구조에서는 edge-use들간의 radial pointer를 각각 저장하고 있기 때문에 이들 포인터들을 따라가면 바로 원하는 다음 face-use로 찾아 갈 수 있으며, 이를 위해 단지 4번의 자료 구조의 필드를

접근하는 시간을 소모하게 된다⁽²⁾. 이에 반하여 본 자료 구조에서는 위의 알고리즘을 사용할 때, 12번의 자료 구조의 필드 접근과 3번의 지역 변수 접근, 7번의 명령 수행을 거쳐서 이웃한 부분 면을 찾아내도록 되어 있다. 이는 Weiler의 소요시간에 비하면 5배 이상의 시간을 요하는 것이나, 본 논문의 방법은 어떠한 기하학적인 계산도 수반하고 있지 않고 단지 위상 요소에 관한 정보만을 이용하기 때문에 여전히 효율적이라고 할 수 있다.

2.4 기억 장소 사용량 비교

본 자료 구조에 의한 기억장소 절감효과를 솔리드 모델에 사용된 Wilson의 방법⁽¹⁴⁾을 원용하여 타 자료 구조와 비교해 보도록 하겠다. 이를 위해서는 먼저 비다양체 모델에 대한 평균적인 위상 요소들의 개수를 면에 대한 함수로 나타내고, 각 위상 요소들이 저장해야 할 최소한의 필드 수를 구한 후, 이들을 각각 곱하여 총계를 계산해 내어야 한다. 그러나 아직까지 비다양체 모델에 대한 평균적인 위상 요소들의 개수를 조사할 자료는 없기 때문에, Table 2에 나타난 다양체 솔리드의 위상 요소들의 평균 개수⁽¹⁴⁾를 사요하였다. 이때 보는 필드들은 포인터를 저장할 수 있는 같은 크기라고 가정하였고, 플래그를 한 비트로 나타낼 수 있으므로 이것의 크기는 무시하였으며, 또한 자료 구조에 나타나는 리스트는 모두 singly linked list로 가정하였으며, 그밖에 꼭 필요하지 않은 필드들은 생략하였다. 각 자료 구조에서 각 위상 요소들이 저장해야 할 최소한의 필드는 ⁽¹⁵⁾에 잘

나타나 있다.

Table 3은 각 자료 구조를 사용해서 표현한 모델의 평균적인 크기를 계산한 결과이다. 본 자료 구조로 표현된 다양체 솔리드 모델은 Radial Edge Structure, Vertex-based B-rep 에 대해 각각 1/2, 1/3정도에 불과하며, 곡면 모델인 경우나 와이어프레임 모델일 경우에도 비슷한 비율이 유지된다. 따라서 솔리드, 곡면, 와이어프레임 모델들이 혼합된 비다양체 모델에 대해서도 Radial Edge Structure, Vertex-based B-rep 에 대해 각각 솔리드 모델의 경우와 유사한 비율로 데이터 감소의 효과가 나타난다고 할 수 있다.

3. 비다양체 모델을 위한 오일러 작업자

3.1 비다양체 모델의 오일러 공식

일반적인 n차원의 오일러-포앙카레 공식(Euler-Poincare formula)은 다음과 같다.

$$\sum_{i=0}^n (-1)^i \alpha_i = \sum_{i=0}^n (-1)^i p_i \tag{1}$$

여기서 n은 차원 수이고, α_i 는 i차원의 단체들의 개수이고, p_i 는 i차원 베티 수(Betti number)⁽¹⁶⁾를 뜻한다. 본 연구의 경계 표현의 대상은 3차원 유클리드 공간에 존재하는 유클리드 포복체로서 이는 점, 선분, 열린 곡면과 닫힌 곡면들의 조합으로 이루어진 모델이므로, 그 표현 영역은 (1)식의 오일러-포앙카레 공식에서 n=2에 해당한다. 따라서 식 (1)은 다음과 같이 된다.

$$V - E + F = P_0 - P_1 + P_2 \tag{2}$$

여기서 좌변의 V, E, F는 각각 꼭지점, 모서리, 면의 개수이며, 우변의 P_0, P_1, P_2 는 각각 0, 1, 2차원 베티 수들이다. 0차원 베티 수 (P_0)는 연결된 도형들의 개수를 나타낸다. 1차원 베티 수(P_1)는 Fig. 12에서와 같이 꼭지점과 모서리로 이루어진 그래프에서 변으로 변환되지 않은 독립적인 회로(cycle)의 개수 및 Fig. 13에서와 같이 곡면 상에서 한 점으로 수축시킬

Table 2. Average number of topological entities in simple solid models of B-rep

Entity	Average
Shells / Object	1
Number of Faces	f
Loops / Face	1
Edges / Loop	6
Number of Edges	3f
Number of Vertices	2f
Edges / Vertex	3

Table 3. Average topological field results for B-rep structures

Data Structure	Number of Record Fields	Model Size of Other Representation / Our Representation
Winged Edge Data Structure	5 + 38f	0.63
Half Edge Data Structure	5 + 44f	0.73
ACIS's Data Structure	9 + 47f	0.78
Radial Edge Structure	14 + 141f	2.4
Vertex-based B-rep	18 + 206f	3.4
Our Representation	8 + 60f	1.0

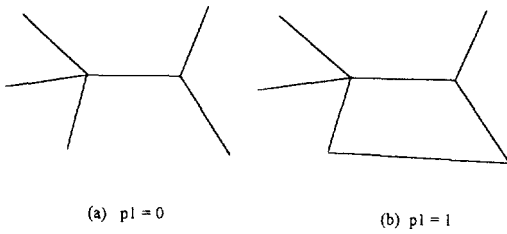


Fig. 12. The first Betti number of wireframe models.

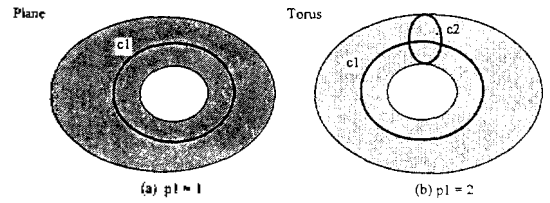
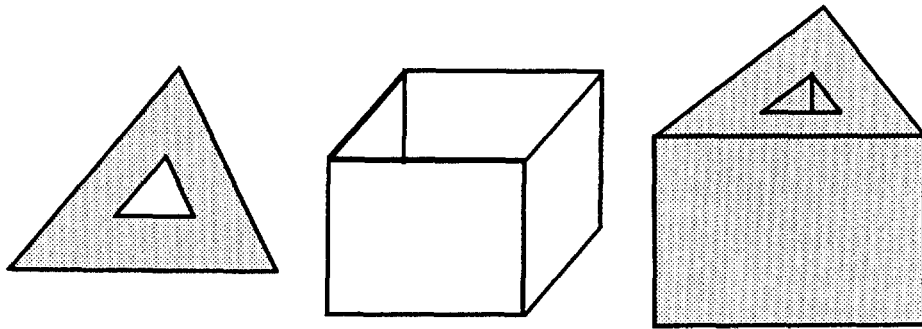


Fig. 13. The first Betti number of surface models.



$v = 6$	$S = 1$	$v = 8$	$S = 1$	$v = 12$	$S = 1$
$e = 6$	$C = 1$	$e = 12$	$C = 0$	$e = 18$	$C = 2$
$f = 1$	$R = 0$	$f = 5$	$R = 0$	$f = 8$	$R = 1$
$l = 1$		$l = 0$		$l = 2$	

Fig. 14. Examples of Euler-Poincare formula for non-manifold models.

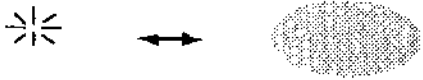
Table 4. Euler operators for non-manifold models

Basic Euler Operators	MEV(KEV)	make(kill) edge, vertex	1 1 0 0 0 0 0
	MEC(KEC)	make(kill) edge, cycle	0 1 0 0 0 1 0
	MFKC(KFMC)	make(kill) face, kill cycle	0 0 1 0 0 -1 0
	MFR(KFR)	make(kill) face, region	0 0 1 0 0 0 1
	MVS(KVS)	make(kill) vertex, shell	1 0 0 0 1 0 0
	MVL(KVL)	make(kill) vertex, loop	1 0 0 1 0 0 0
Additional Euler Operators	SEMV(JEKV)	split(join) edge, make(kill) vertex	1 1 0 0 0 0 0
	MEF(KEF)	make(kill) edge, face	0 1 0 0 1 0 0
	KEML(MEKL)	kill(make) edge, make(kill) loop	0 -1 0 1 0 0 0
	KEMS(MEKS)	kill(make) edge, make(kill) shell	0 -1 0 0 1 0 0
Additional Topological Operators	MMR(KMR)	make(kill) model, region	

수 없는 원의 개수를 나타낸다. 2차원 베타수(P_2)는 이어진 면들에 의해 만들어진 영역의 수이다.

이 식의 각 항을 본 표현에서 사용하는 위상 요소들로 나타내 보면, P_0 는 안쪽 셀의 개수(S), P_1 는 무한대 영역을 제외한 나머지 영역의 개수(R)를 뜻하며, P_2 는 위상 요소로 나타나지 않으나 이를 최로의

개수 C 로 나타내면 식(2)의 우변은 $S-C+R$ 이 된다. 한편, 면의 내부에 안쪽 루프가 L 개 존재하는 경우, L 개의 모서리를 추가하여 이를 제거하면 (2)식의 원래의 E 는 $E+L$ 로 증가하게 되며, 이를 (2)식의 좌변에 대입하면 이는 $V-(E+L)+F$ 즉, $V-E+F-L$ 이 된다. 따라서 위 (2)식은 다음과 같이 된다.



(a) MMR (KMR)



(g) MVL (KVL)



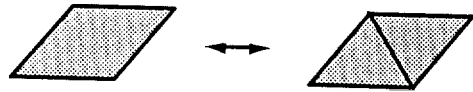
(b) MVS (KVS)



(h) SEMV (JEKV)



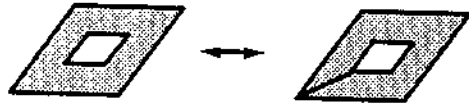
(c) MEV (KEV)



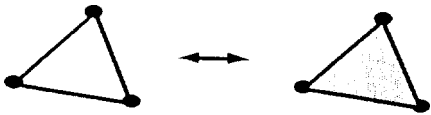
(i) MEF (KEF)



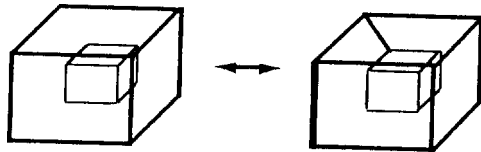
(d) MEC (KEC)



(j) MEKL (KEML)



(e) MFKC (KFMC)



(k) MEKS (KEMS)



(f) MFR (KFR)

Fig. 15. Euler operators for non-manifold models.

$$V - E + F - L = S - C + R \quad (3)$$

이 식이 본 시스템에서 사용하고 있는 오일러 작업자의 바탕이 되는 오일러 공식이며, 이 식으로부터 솔리드 모델과 r-set 모델에 대한 기존의 오일러 공식을 유도해 내는 것이 가능하다⁽¹⁴⁾. Fig. 14에서는 몇 가지 비다양체 모델에 대하여 식(3)이 만족되고 있음을 보여주고 있다.

3.2 오일러 작업자

비다양체 모델에 대한 오일러 공식인 식(3)은 7개의 변수로 구성되어 있으므로 이들로 이루어진 벡터 공간 (V, E, F, L, S, C, R)을 생각하면, 독립적인 기저 벡터(basis vector)는 6개이다. 따라서 비다양체 모델에서 이론적으로 필요한 최소한의 오일러 작업자는 역작업자를 포함해서 총 12개가 된다. 본 논문에서는 이 12개로 구성된 기본적인 오일러 작업자들을 선택하였다. 그러나 이것만으로는 모델링을 충분히 효율적으로 구현할 수 없으므로 이에 덧붙여 8개의 오일러 작업자를 추가시켰다. 또한 이 오일러 작업자 이외에, 최초로 모델과 무한대 영역을 생성시키는 MMR (Make Model, Region)과 마지막으로 모델과 무한대 영역을 지우는 KMR(Kill Model, Region) 두 위상 작업자들을 추가하였다. 이러한 총 20개의 오일러 작업자과 2개의 기본적인 위상 작업자에 대한 리스트가 Table 4에 나타나 있으며, 그 각각에 대한 작업 내용은 Fig. 15에 간략히 그림으로 설명되어 있다. 또한 각 오일러 작업자의 함수 및 입출력 변수에 대한 정의가 부록 C에 나타나 있다. 여기에 사용된 오일러 작업자들의 이름은 관례에 따라 다음과 같이 암호화된 약어를 써서 나타내었다.

- M - make, K - kill, S - split, J - join
- V - vertex, E - edge, F - face, L - hole loop
- S - void shell, C - cycle, R - region

이 위상 작업자들과 오일러 작업자들은 기본적인 입체 형상의 생성, 스위핑(sweeping)작업 및 불리안 작업과 같은 상위 레벨의 모델링 작업들을 구현하는데 사용된다.

4. 결 론

본 논문에서는 효율성을 유지하면서 중복되는 정보를 최대한 줄일 수 있는 곡면 경계 정보를 중심으로 한 간결한 형태의 계층 구조적인 비다양체 모델

에 대한 자료 구조를 제안하고, 또한 오일러 공식을 바탕으로 비다양체 모델의 오일러 작업자들을 구현하였다. 이와 같이 이해하기 쉬운 간결한 형태의 자료 구조를 제안함으로써, 지금까지의 비다양체 모델에 대한 자료 구조들이 가지고 있던 위상 정보의 중복으로 인한 기억 저장 장소의 낭비를 제거할 수 있으며, 이를 바탕으로 한 자료구조 관리 프로그램의 개발을 보다 쉽게 해줄 수 있게 되었다. 또한 비다양체 모델을 생성, 수정할 수 있는 오일러 작업자들에 대한 사양을 제시하고 이를 구현함으로써, 모델링 과정에서 모델에 대한 위상적인 정합성을 유지시키고, 특정한 자료 구조로부터 상위 레벨의 모델링 작업들을 절연시키며, 또한 실행한 오일러 작업자에 대한 역작업자를 사용하여 취소 명령을 쉽게 구현할 수 있도록 하였다.

참고문헌

1. Masuda, H., "Topological operators and Boolean operations for complex-based non-manifold geometric models," *Computer-Aided Design*, Vol.25, No.2, pp.119-129, Feb. 1993.
2. Weiler, K., "Topological Structures for Geometric Modeling," *PHD. Thesis*, Rensselaer Polytechnic Institute, U.S.A., Aug. 1986.
3. Crocker, G.A. and Reinke, W.F., "An Editable Non-manifold Boundary Representation," *IEEE Computer Graphics & Applications*, Vol.11, No.3, Mar. 1991.
4. Masuda, H., "Form-Feature Representation based on Non-manifold Geometric Modeling," *Proceedings of MICAD'92*, 1992.
5. 김성환, "방향 설계 방식의 조립체 모델링을 지원하는 특징 형상 기반의 비다양체 CAD 시스템의 개발," 박사 학위 논문, 서울 대학교, 1994.
6. Mantyla, M., *An Introduction to Solid Modeling*, Computer Science Press, 1988.
7. Choi, Y., "Vertex-based Boundary Representation of Non-manifold Geometric Models," *PhD. Thesis*, Carnegie Mellon Univ., U.S.A., Aug. 1989.
8. Yamaguchi, Y. and Kimura, F., "Nonmanifold Topology Based on Coupling Entities," *IEEE Computer Graphics and Applications*, Vol.15, No.1, pp. 42-50, Jan. 1995.
9. Rossignac, J. and O'Conner, M.A., "SGC: A Dimension-independent Model for Pointsets with Internal Structures and Incomplete Boundaries," *Geometric Modeling for Product Engineering*, North-Holland, pp.145-180, 1990.

10. Luo, Y., "Generalized Euler Operators for Non-Manifold Boundary Solid Modeling," *Geometric Modeling Studies 1990/3, MTA SZTAKI*, Hungary, pp.19-34, 1993.
11. Heisserman, J.A., "A Generalized Euler-Poincare Equation," *Symposium on Solid Modeling Foundations and CAD/CAM Applications*, Austin, Texas, Jun. 1991.
12. Spatial Technology Inc., *ACIS Geometric Modeler, Interface Guide*, Spatial Technology Inc., U.S.A., 1991.
13. 이상현, "사출 성형 제품의 설계 및 해석의 통합 환경을 제공하기 위한 특징 형상 기반 미다양체 모델링 시스템의 개발," 박사 학위 논문, 서울 대학교, 1993.
14. Wilson, P.R., "Data Transfer and Solid Modeling," *Geometric Modeling for CAD Applications*, Elsevier Science Publishers B.V., North-Holland, pp.217-254, 1988.



이 상 현

1986년 서울대학교 기계설계학과 학사
 1988년 서울대학교 기계설계학과 석사
 1993년 서울대학교 기계설계학과 박사
 1993년 ~ 1995년 (주)신도리코 기술연구소
 1996년 ~ 현재 교정기술연구원 생산기술연
 구실 선임연구원
 관심분야: Non-manifold Modeling, Intelligent
 CAD, Rapid prototyping



이 견 우

1978년 서울대학교 기계공학과 학사
 1981년 미국 MIT 공학석사
 1984년 미국 MIT 공학박사
 1984년 ~ 1986년 Univ. of Illinois at Urbana-
 Champaign 조교수
 1986년 ~ 현재 서울대학교 기계설계학과
 교수
 관심분야: Computer aided geometric
 design, NC tool path generation
 and verification, Solid modeling,
 Rapid prototyping

<부 록>

A. 자료 구조

```
class Entity {
    int      id;
    Attribute *attrib;
};
class Model : public Entity {
    Model    *_next;
    Region   *_region;    // list of regions in this model
    Face     *_face;     // list of faces
    Edge     *_edge;     // list of edges
    Vertex   *_vertex;   // list of vertices
};
class Region : public Entity {
    Region   *_next;     // link field of region list of model
    Model    *_model;   // parent model
    Shell    *_shell;   // peripheral shell of region
};
class Shell : public Entity {
    Shell    *_next;    // next void shell
    Region   *_region; // parent region
    Pface    *_pface;  // partial face
};
class Pface : public Entity { // partial face(p-face)
    Pface    *_next;    // next partial face
    Shell    *_shell;   // parent shell
    Entity   *_child;  // child entity: one of face, edge or vertex
    Orient   _orient;  // orientation flag w.r.t. face normal
    Pface    *_mate;   // mate partial face
};
```



```

};
class Face : public Entity {
    Face    *_next;           //link field of face list of model
    Pface   *_pface;        //peripheral loop of face
    Loop    *_loop;         //peripheral loop of face
    Surface *_geometry;     //surface of face
};
class Loop : public Entity {
    Loop    *_next;         //next hole loop
    Face    *_face;        //parent face
    Pedge   *_pedge;       //partial edge
};
class Pedge : public Entity { //partial edge(p-edge)
    Loop    *_loop;        //parent loop
    Entity  *_child;       //child entity: one of edge or-vertex
    Orient  *_orient;      //orientation flag w.r.t edge direction
    Pedge   *_looped_prev; //previous p-edge in loop
    Pedge   *_looped_next; //next p-edge in loop
    Pedge   *_radial_prev; //previous p-edge in radial cycle of edge
    Pedge   *_radial_next; //next p-edge in radial cycle of edge
};
class Edge : public Entity {
    Edge    *_next;        //link field of edge list of model
    Entity  *_parent;      //parent entity: one of p-edge or p-face
    Pvertex *_pvertex{2}; //two end partial vertices of edge
    Curve   *_geometry;    //curve of edge
};
class Pvertex : public Entity { //partial vertex(p-vertex)
    Pvertex *_next;        //next p-vertex of the same vertex
    Entity  *_parent;      //parent entity: one of edge or p-edge
    Vertex  *_vertex;     //vertex from which p-vertex is originated
};
class Vertex : public Entity {
    Vertex  *_next;       //link field of vertex list of model
    Entity  *_parent;     //parent entity: one of p-vertex or p-face
    Point   *_geomerty;  //position of vertex
};
};

```

B. 인접한 부분 면 찾기 알고리즘

```

//
// Declare the class for virtual edge-use
//
class VirtualEdgeUse {
    Pedge  *_partial_edge; //pointer of partial edge
    Orient orient;         //orientation w.r.t. partial_edge
public:
    :
};
// -----
// Function: searches for the next partial face in radial cycle
// Input: pf1 - current partial face

```

```

//      pe1 - current partial edge on pf1
// Output: pf3 - next adjacent partial face
//      pe2 - next adjacent partial edge on next_pface
//-----
void get_radial_next_partial_face(Pface *pf1, Pedge *pe1, Pface *pf3, Pedge *pe3)
    • VirtualEdgeUse eu2;
      //
      // -- (Step 1& 2) Set virtual edge-use(eu2) that is the mate of 'eu1'
      //
    • eu2.set_partial_edge(pe1);
    • if(pf1->orient( ) == SAME)      // if the orientation of pf1 is the same
                                      // as that of pf1's face
        eu2.set_orient(REVERSE);
    • else
        eu2.set_orient(SAME);
      //
      // -- (Step 3) Determine the next partial edge(pe2) in radial cycle
      //
    • if(eu2.start_vertex( ) == pe1->edge( )->start_vertex( ))
        pe2 = pe1->radial_prev( );
    • else
        pe2 = pe1->radial_next( );
      //
      // -- (Step 4) Determine the next partial face(pf3)
      //
    • if(pe2->start_vertex( ) == eu2.start_vertex( ))
        ⊙ pf3 = pe2->face( )->partial_face_of_same_normal( );
    • else
        ⊙ pf3 = pe2->face( )->partial_face_of_opposite_normal( );

```

C. 오일러 작업자

각 오일러 작업자에 대한 함수에서 입력 변수는 위상 요소 또는 기하학적 요소에 대한 싱글 포인터를, 출력 변수는 더블 포인터를 사용하였다.

C.1 기본적인 오일러 작업자

MEV (Vertex *v, Entity *parent, Edge **newe, Vertex **newv, Curve *cv)

KEV (Edge *e, Vertex *v)

MEC (Shell *s, Vertex *v1, Vertex *v2, Edge **newe, Curve *cv)

KEC (Edge *e)

MFKC (Shell *s, EdgeList elist, Face **newf, Surface *sf)

KFMC (Face *f)

MFR (Shell *s, EdgeList elist, Face **newf, Region **newr, Surface *sf)

KFR (Face *f, Region *r)

MVS (Region *r, Vertex **newv, Shell **news, Point *pt)

KVS (Shell *s)

MVL (Face *f, Vertex **v, Loop **newl, Point *pt)

KVL (Loop *l)

C.2 추가적인 오일러 작업자

SEMV (Edge *e, Edge **newe, Vertex **newv, Point *pt)

JEKV (Edge *e, Vertex *v)

MEF (Loop *, Vertex *v1, Vertex *v2, Edge **e, Face **f, Curve *cv)

KEF (Edge *e, Face *f)

KEML (Edge *e, Loop **l)

MEKL (Loop *l1, Loop *l2, Vertex *v1, Vertex *v2, Edge **newe, Curve *cv)

KEMS (Edge *e, Shell **news)

MEKS (Shell *s1, Shell *s2, Vertex *v1, Vertex *v2, Edge **newe, Curve *cv)

C.3 기타 위상 작업자

MMR (Model **newm, Region **newr)

KMR (Model *m)