

<기술논문>

분해 모델을 이용한 자동차 차체의 틈새 탐색 시스템 개발

배 원 중¹⁾ · 이 승 훈¹⁾ · 박 성 배²⁾ · 정 용 호³⁾

부산대학교 대학원 기계공학부¹⁾ · 현대자동차 차체생기 1팀²⁾ · 부산대학교 기계공학부³⁾

Development of Gap Searching System for Car Body Assembly by Decomposition Model Representation

Wonjung Bae¹⁾ · Sunghoon Lee¹⁾ · Sungbae Park²⁾ · Yoongho Jung³⁾

¹⁾Graduate School of Mechanical Engineering, Pusan National University, Busan 609-735, Korea

²⁾Body Manufacturing Engineering Team 1, Hyundai Motor Co., 700 Yangchung-dong, Buk-gu, Ulsan 683-791, Korea

³⁾School of Mechanical Engineering, Pusan National University, Busan 609-735, Korea

(Received 4 October 2011 / Revised 24 November 2011 / Accepted 7 December 2011)

Abstract : Large number of part design for aircraft and automobile is preceded by functional or sectional design groups for efficiency. However, interferences and gaps can be found when the parts and sub-assemblies by those design groups are to be assembled. These interferences and gaps cause design changes and additional repair processes. While interference problem has been resolved by digital mockup and concurrent engineering methodology, gap problem has been covered by temporary treatment of filling gap with sealant. This kind of fast fix causes fatal problem of leakage when the gap is too big for filling or the treatment gets old. With this research, we have developed a program to find the gap automatically among parts of assembly so that users can find them to correct their design before manufacturing stage. By using decomposition model representation method, the developed program can search the gap among complex car body parts to be visualized with volumetric information. It can also define the boundary between the gap and exterior empty space automatically. Though we have proved the efficiency of the developed program by applying to automobile assembly, application of the program is not limited to car body only, but also can be extended to aircraft and ship design of large number of parts.

Key words : Mismatch(불일치), Octree(옥트리), Collaborative work(협업 작업), Virtual assembly(가상 조립), Automotive(자동차)

1. 서 론

자동차, 항공기, 선박과 같이 부품수가 많은 대형 조립체 개발 과정의 특징은 개발 기간이 길며 다수의 설계자가 동시에 설계를 한다는 것이다. 이러한 대형 조립체의 개발에서 부품간의 간섭과 충돌, 완전히 결합 되지 않는 틈새(gap) 등은 제작 및 조립 단계에서 설계 변경의 주요 원인으로 작용하는데, 이는 최초에 계획한 개발 기간과 개발 비용을 증가시

키게 된다. 부품간의 간섭이나 충돌의 경우, 종래에는 부품을 제작하기 전에 실물 mock-up을 제작하여 이를 검증하였고, 최근에는 동시 공학적 설계(concurrent engineering)¹⁾ 개념을 도입하여, 삼차원의 솔리드 모델로 설계된 각각의 부품을 컴퓨터로 가상 조립하는 “digital mockup(DMU)” 시스템²⁾을 통하여 설계 단계에서 부품의 제작성과 조립성을 확인하여 개발 기간을 단축하고 생산성을 높이는 방법이 적용되고 있다.

한편, 서로 접하는 부품들이 정확히 일치하지 않

*Corresponding author, E-mail: yhj@pusan.ac.kr

는 틈새의 경우에도 많은 문제를 야기할 수 있다. 특히 자동차의 경우, 차체를 구성하는 부품들이 얇은 판재로 조립 또는 용접되는데 이러한 부품들 사이에 의도하지 않은 틈새가 존재한 상태로 조립되면 틈새 사이로 빗물이나 오수가 스며들 수 있다. 이러한 누수는 세균의 번식과 부품의 부식, 누수에 의한 감전 등 여러 가지 문제점들을 야기해 부품의 내구성과 상품 가치에 치명적인 악영향을 미치게 된다. 이러한 틈새는 제작 과정에서 판재의 스프링 백(Spring back) 등의 원인이나 단순한 설계 오류로 야기될 수도 있으나, 과도한 틈새는 대부분 서로 다른 설계자들이 설계한 부품들이 서로 접합되는 곳을 확인하지 못함에 기인한다. 이러한 문제를 해결하기 위해 기존의 방법에서는 설계 단계에서 조립체의 단면도를 사용하여 틈새를 찾아내거나, 조립 후 과도한 틈새를 발견하였을 때 밀폐제로 충전하는 임시적인 방법을 사용하고 있다. 그러나 설계 단계에서 단면도로써 틈새를 확인하는 방법은 시간이 많이 소요되고 틈새의 확인이 불완전하기 때문에 보다 근본적인 해결 방법이 모색되어야 한다.

부품간의 간섭이나 충돌의 경우를 설계 단계에서 탐색하고 검증하는 방법들에 대해 이미 많은 연구가 이루어져 왔다. 가상공간에서 조립체의 조립 검사를 수행한 연구로서 Song과 Chung³⁾은 XML을 이용하여 CAD 데이터의 조립체의 구성과 효율적인 가시화 데이터 생성에 응용되는 DMU 시스템을 제안하였으며, PDM(Product Data Management)을 기반으로 인터넷 환경에서의 대용량 조립체의 CAD 데이터의 검토 작업, 실시간 간섭체크, 디자인 검증 작업등이 수행 가능한 DMU 시스템도 구축하였다. Park 등⁴⁾은 DMU 뷰어 시스템 상에서 해석이 가능한 복합 간섭 검사 알고리즘을 제안하였다. Chen⁵⁾는 지역적으로 떨어진 설계자들이 인터넷 상에서 실시간으로 조립체를 설계하는 시스템과, 이를 위한 조립체의 표현 방법 및 설계 조건을 제안하였으며, Chen⁶⁾은 부품을 가상 조립하고 부품 간의 충돌을 실시간으로 탐지하는 시스템을 개발하였다.

그러나 본 연구에서 목적하는 바와 같은 부품 간의 틈새를 자동으로 탐색하는 방법에 대한 연구는 수행되지 않았다. 본 연구에서는 복잡한 형상의 부품들이 조립 될 때 부품들 사이의 틈새를 자동으로

탐색하고 이를 가시화함으로써 설계자가 설계 단계에서 과도한 틈새를 미리 파악하여 대처할 수 있는 방법과 도구를 개발 하고자 한다.

2. 틈새 탐색 방법의 개요

본 연구에서는 복잡한 형상의 부품들로 조립 또는 용접되는 조립체에서 틈새를 자동으로 탐색하고 이를 가시화 하는 방법을 개발하고자 한다. 특히 자동차, 항공기, 선박 등에 사용되는 판재는 대부분 자유 곡면을 포함하고 있으므로 본 연구에서는 가장 포괄적인 형상인 자유 곡면 형상의 판재까지를 대상으로 하겠다. 이 경우, 부품 사이에 발생하는 틈새도 자유 곡면을 포함하는 복잡한 경계면을 가진 영역이 될 수 있는데, 이를 자동으로 탐색하고 가시화하기 위해 주어진 형상의 복잡성에 제한을 받지 않으며 계산의 효율을 높일 수 있는 ‘분해’(decomposition) 모델로 표현하고자 한다. 분해 모델의 표현 방법에는 대표적으로 균일한 크기의 복셀(voxel)^{7,8)}로 표현하는 방법과 매수 배의 크기의 요소로 표현하는 옥트리(octree)^{9,10)} 방법이 있는데, 본 연구에서는 이 두 가지 방법을 혼합한 방법¹¹⁾을 사용하며, 전체적인 알고리즘을 Fig. 1에 나타내었다. 먼저 틈새를 탐색하고자 하는 조립체의 각 부품을 STL(Stereolithographic Language) 파일 형식으로 입력 받는다. 본 연구의 대상은 부품수가 많은 조립체의 경우이므로, 모델의 데이터가 크면 개발되는 프로그램의 실행 속도가 문제가 될 수 있다. 또한 다른 종류의 CAD시스템을 사용하는 경우에도 개발되는 프로그램이 적용될 수 있기 위해 CAD모델의 표준 데이터 형식으로 널리 채택되고 있고 데이터 크기가 작은 STL 파일 형식으로 부품을 입력받는다.

그 다음, 틈새를 탐색할 영역 내의 각각의 부품을 주어진 해상도까지 옥트리로 분할한다. 이때 각 부품들이 조립된 상태를 나타내기 위해 전체 조립체에 대한 옥트리로 표현한 다음, 이를 옥트리의 최소 요소 크기의 복셀로 분할한다. 복셀로 표현된 모든 단품들을 각자의 위치에 배치하여 전체 조립체의 분해 모델을 완성한 후, 틈새 영역에 해당하는 복셀들을 탐색하고 이를 가시화 한다. 다음 장에서 각 단계에 대해 보다 자세히 설명된다.

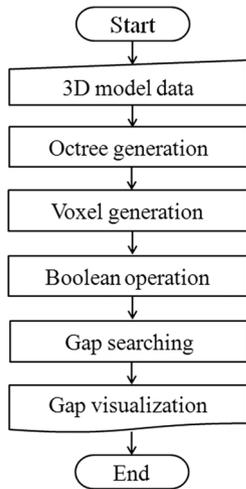


Fig. 1 Gap visualization algorithm

3. 조립체에 대한 분해 모델의 생성

본 연구에서의 틈새란 조립체에서 부품과 부품 사이의 공간으로 정의할 수 있는데, 부품들의 형상이 복잡하면 틈새의 형상 또한 복잡하게 된다. 본 연구에서는 이를 자동으로 탐색하고 가시화하기 위해 공간과 부품을 ‘분해’ 모델로 표현한다. 즉 부품들이 차지하는 공간을 분해 모델로 표현해 두면, 3차원 공간에서 이를 제외한 공간이 부품의 외부, 또는 부품들 사이의 공간인 틈새 영역이 될 수 있다. 이를 위해 각 단품에 대한 분해 모델을 먼저 생성한 후, 이들을 부울리안(boolean) 연산으로 합집합을 구하여 전체 조립체에 대한 분해 모델을 생성한다.

3.1 단품의 옥트리 모델 생성

입체 형상을 분해 모델로 표현하는 대표적인 방법에는 크기가 균일한 복셀로 표현하는 방법과 배수 배의 크기를 가진 요소들로 분할하는 옥트리 방법이 있다. 복셀로 표현하는 방법은 알고리즘이 간단하지만, 복셀의 크기가 작아질수록 복셀의 개수가 급격히 증가하여 데이터의 크기가 급격히 증가한다. 또한 각각의 복셀이 주어진 물체의 내부에 존재하는지를 판단하는데 소요되는 시간이 복셀의 개수에 비례하여 증가하는 단점이 있다.

반면 옥트리는 Fig. 2와 같이 물체를 포함하는 최소한의 육면체인 ‘뿌리’ 요소(root octant)를 구하고

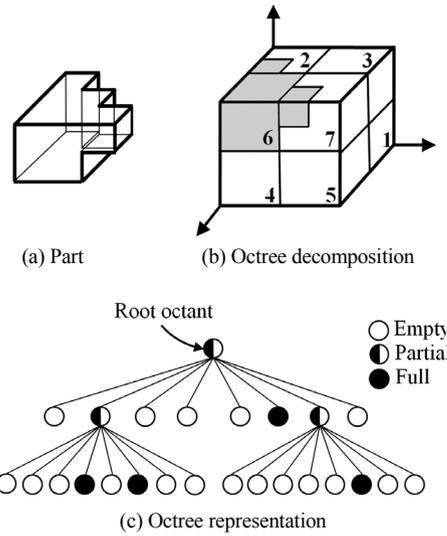
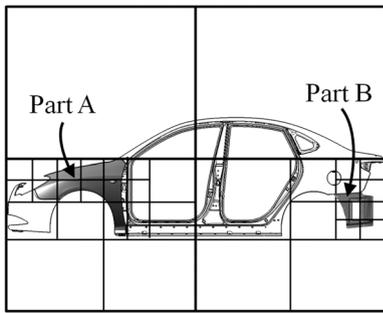


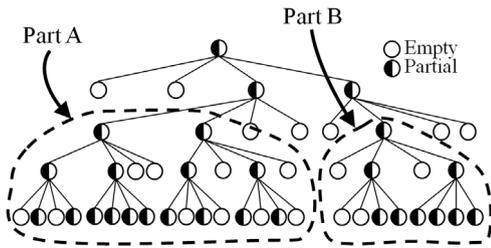
Fig. 2 Octree representation

이를 여덟 개의 ‘자식’(octant)으로 분할한 다음, 각각의 ‘자식’과 대상 물체와의 포함 관계를 판별하여 ‘내부’(full), ‘외부’(empty), ‘부분’(partial) 요소로 분류한다. 그 다음, ‘부분’으로 분류된 요소는 다시 여덟개의 자식들로 분할하는 앞의 과정을 반복함으로써 크기가 다른 요소들로 물체를 근사화 할 수 있다. 옥트리 방법은 물체의 경계를 포함하는 ‘부분’ 요소를 주어진 해상도 수준까지 계속 분할하기 때문에 복셀에 비해 요소의 개수와 전체적인 계산 시간을 훨씬 줄일 수 있는 이점이 있다. 본 연구에서는 이 두 방법의 장점을 혼합한 방법을 사용하고자 한다.

본 연구의 대상은 조립체의 경우인데, 입력으로 받는 STL 파일은 각각의 단품을 나타내며 전체 조립체를 나타낼 수 없다. 따라서 전체 조립체를 분해 모델로 나타내기 위해서는 각각의 부품을 옥트리로 나타낸 후, 이들을 부울리안 연산에 의해 합쳐야 한다. 이를 위해 각 부품을 옥트리로 나타낼 때, 각 부품을 포함하는 최소한의 ‘뿌리’ 요소를 대상으로 옥트리를 생성하는 것이 아니라, Fig. 3과 같이 전체 조립체를 포함하는 ‘뿌리’ 요소를 대상으로 옥트리를 생성한다. Fig. 3은 이해의 편의를 위해 이차원의 쿼드트리(quadtree)로 나타내었다. Fig. 3(a)에서 특정 부품 A (Part A)에 대한 쿼드트리의 표현은 Fig. 3(b) 왼쪽의 점선 내부의 쿼드트리 구조만으로 표현할 수 있지만, 본 연구에서는 전체 조립체를 나타내기



(a) Quadtree subdivision for assembly



(b) Quadtree structure for assemble

Fig. 3 Quadtree representation for assembly

위해 Fig. 3(b)와 같이 전체 조립체에 대한 쿼드트리로 표현한다.

마찬가지로 다른 부품 B (Part B)의 경우도 Fig. 3(b)와 같이 전체 조립체에 대한 쿼드트리로 표현하고 앞에서 생성된 부품 A의 쿼드트리와 합성하여 조립된 상태를 표현한다. 마찬가지로 다른 부품도 전체 조립체의 쿼드트리에 합성하면 모든 부품이 조립된 상태를 표현할 수 있다.

한편, 옥트리의 생성과정에서 각 요소가 물체의 ‘내부’의 요소인지 ‘외부’의 요소인지를 판단하기 위한 계산 과정이 많은 시간을 요한다. 본 연구에서는 이러한 계산 시간을 단축하기 위해 옥트리를 생성하는 단계에서는 ‘내부’ 요소와 ‘외부’ 요소를 구분하지 않고, 다음 장에서 설명할 틈새의 탐색 단계에서 ‘외부’ 요소와 틈새에 해당하는 요소로 구분한다.

3.2 옥트리 생성 후 복셀 모델의 생성

옥트리는 물체의 형상이나 공간을 분해모델로 표현하는 효과적이고 구조적인 방법이다. 그러나 복셀은 틈새에 해당하는 영역과 외부 공간과의 경계면을 보다 매끄럽게 표현할 수 있다. 또한 틈새에 해당하는 영역을 탐색하기 위해서는 이웃하는 요소를

찾는 과정이 매우 빈번한데, 옥트리의 경우 이웃 찾기(neighbor finding) 알고리즘^{12,13)}을 사용할 수 있으나 복셀의 경우보다 많은 시간을 요한다. 그러나 처음부터 복셀 모델로 나타내는 방법은 모든 복셀 요소와 대상으로 하는 부품과의 포함 관계를 판단하는데 시간이 너무 많이 소요되는 단점이 있다. 본 연구에서는 이 두 방법의 장점을 취하기 위해 주어진 부품을 옥트리로 먼저 표현한 후, 옥트리로 표현된 부품을 포함하는 최소 크기의 육면체에 대해 옥트리의 최소 요소의 크기로 균일하게 분할한다. 이러한 방법은 옥트리의 ‘내부’ 요소 또는 ‘외부’ 요소로부터 분할되는 복셀은 대상 물체와의 포함 관계를 따로 계산할 필요가 없기 때문에 계산 시간을 훨씬 줄일 수 있다.

이해를 돕기 위해 Fig. 4에 이차원의 경우를 예로 나타내었는데, Fig. 4의 위쪽 그림과 같이 전체 조립체를 포함하는 ‘뿌리’ 요소로부터 시작하여 대상이 되는 부품에 대한 쿼드트리를 생성한다. 그 다음, Fig. 4의 아래쪽 그림과 같이 주어진 부품의 ‘부분’ 요소들을 포함하는 최소 크기의 사각형을 계산하고, 이를 최소 크기의 요소로 균일하게 분할한다. 분할된 각 요소는 해당 부품의 고유번호와 그 부품과 분해 모델을 나타내기 위해 본 연구에서 옥트리와 복셀의 혼합한 방법을 채택한 이유는 다음과 같은

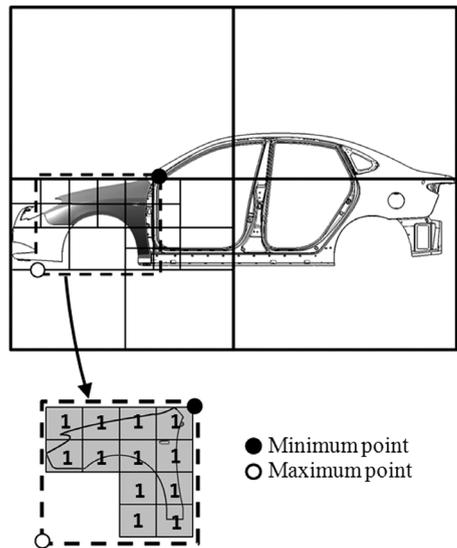


Fig. 4 Decomposition model from quadtree representation

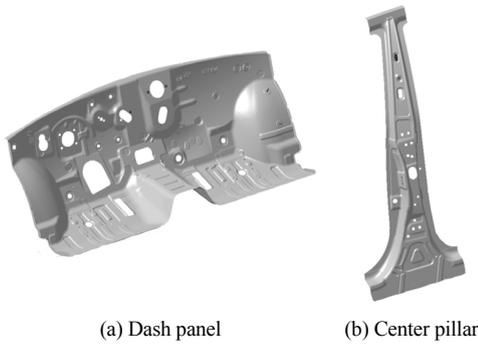


Fig. 5 Sample models for comparison

세 가지이다. 첫째, 본 연구에서 채택한 방법은 복셀의 장점을 그대로 가지지만 처음부터 균일한 크기의 복셀로 나타내는 방법에 비해 계산 시간이 훨씬 적게 소요된다. 왜냐하면 옥트리의 ‘내부’ 또는 ‘외부’ 요소를 분할하여 생성된 복셀들은 모두 해당하는 옥트리 요소의 포함 관계와 같기 때문에 대상 부품에 대한 포함 관계를 다시 계산할 필요가 없다. 이를 비교하기 위해 Fig. 5와 같은 얇은 판재 형상인 자동차 부품에 대해 옥트리로 나타낸 경우와, 본 연구에서 채택한 방법인 옥트리로 나타낸 이후 복셀로 나타낸 경우 및 처음부터 복셀로 나타낸 경우의 처리 속도를 아래의 Table 1에 나타내었다. Table 1의 결과는 Intel Core i7의 CPU 2.94GHz, RAM 3.0 GB의 PC에서 계산한 결과인데, Fig. 5의 모델들의 경우 본 연구에서 채택한 방법이 옥트리로 나타내는 방법보다는 다소 느리지만, 처음부터 복셀로 나타내는 방법에 비해 평균 70배 정도 빠른 것을 알 수 있다.

둘째, 분해 모델의 표현을 위해 최종적으로 복셀로 근사화하는 또 다른 이유는, 동일한 빈 공간이지만 틈새에 해당하는 공간과 물체의 외부에 해당하는 공간을 자동으로 구분하는 알고리즘을 적용하기 위해서이다. 틈새를 탐색하기 위해서는 틈새에 해당하는 공간만이 폐 영역(closed volume)으로 정의

Table 1 Decomposition model generation time (sec)

	Dash panel (65241 triangles)	Center pillar (18575 triangles)
Octree	2	1
Octree+Voxel*	3	2
Voxel	212	133

* This research

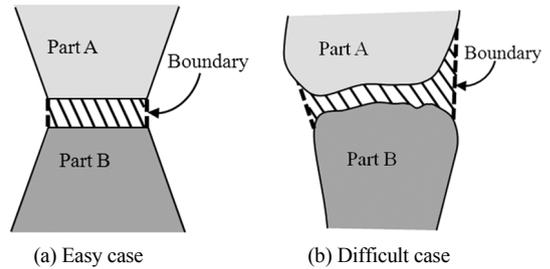


Fig. 6 Boundary definition for gap region

되어야 하는데, 예를 들어 Fig. 6(a)와 같은 경우는 두 부품 사이의 틈새와 외부 공간을 구분하기가 용이하지만 Fig. 6(b)와 같이 자유 곡면의 형상을 가진 부품의 경우, 틈새 영역과 외부 공간을 구분하는 경계면(점선으로 표시)의 위치를 정의하기가 모호한 문제점이 있다. 본 연구에서는 부품과 공간이 균일한 크기의 복셀로 표시되어 있을 때, 이들을 구분하여 틈새에 해당하는 영역을 탐색하는 알고리즘을 개발하였으며, 이는 다음 장에서 자세히 설명된다.

셋째, 만약 부품이 옥트리로 표현되어 있을 경우, 틈새 영역의 경계면을 표현하는 요소들도 다양한 크기를 가질 수 있어 경계면이 매끄럽지 않게 표현될 수 있지만, 균일한 크기의 복셀은 경계면을 보다 매끄럽게 나타낼 수 있다.

3.3 조립체에 대한 분해 모델의 생성

각각의 부품들을 최종적으로 복셀로 나타낸 후, 모든 부품들이 조립된 상태를 나타내기 위해 각 부품들의 복셀 모델을 Fig. 7과 같이 부울리안 연산에 의해 합친다. 즉 Fig. 7은 이해를 돕기 위해 두 개의 부품이 조립되는 상태를 이차원의 경우에 대해 나타내었는데, Fig. 7(a)와 같이 이전의 단계에서 ‘Part 1’과 ‘Part 2’를 각각 포함하는 최소 크기의 사각형을 이미 생성해 두었으므로, 이를 이용하여 Fig. 7(b)와 같이 두 부품의 합집합을 포함하는 최소 크기의 사각형을 구한다. 그 다음, Fig. 7(c)와 같이 합집합의 사각형을 복셀로 분할한 후, ‘Part 1’과 ‘Part 2’의 복셀을 합집합의 해당 위치로 매핑(mapping)한다. 이때, 합집합의 각 복셀은 각 부품의 원래 복셀이 가지고 있던 정보를 그대로 복사하여 가진다. 한편, Fig. 7(c)에서 ‘E 구역’ 내에는 하나의 복셀에 다수의 부품이 겹쳐져 있는 경우가 있는데, 그 복셀은 포함된 모든

요소이므로 일곱 번째 요소부터 아홉 번째 요소(빗금친 요소)들을 ‘틈새’ 요소로 분류한다.

이러한 과정을 조립체 복셀 모델의 최하위 행부터 최상위 행까지 진행한다. 마찬가지로 방법으로 맨 왼쪽 열부터 맨 오른쪽 열까지 y-축 방향으로 탐색하여 Fig. 8(b)와 같이 ‘틈새’ (빗금친 요소)로 분류된 모든 요소를 구한다. 실제의 경우는 앞의 과정에서 z-축 방향의 탐색을 추가하면 삼차원의 틈새 영역을 모두 찾을 수 있다. 제안된 방법은 틈새 영역과 외부 영역의 경계면을 자동으로 구분하는 장점도 있다.

4.2 틈새 영역의 가시화

제안된 틈새 탐색 알고리즘으로 부품 사이의 공간 영역에 속하는 요소를 모두 찾아 이들을 구분하여 나타냄으로써 틈새 영역을 가시화할 수 있다. 그런데, 앞의 소절에서 설명한 틈새 탐색 알고리즘은 Fig. 7(c)의 ‘E 영역’과 같은 경우에 대해 틈새가 없다고 판단하였다. 즉, Fig. 9(a)는 ‘E 영역’만을 나타내었는데, Fig. 9(b)와 같이 이들은 서로 다른 부품에 속해 있는 ‘부분’ 요소를 사이에 빈 공간이 없기 때문에 틈새 요소가 없는 것으로 판단하였었다. 그러나 실제로는 이들 부품들 사이에서도 틈새가 존재하므로 이들 공간도 틈새로 분류되고 가시화되어야 한다.

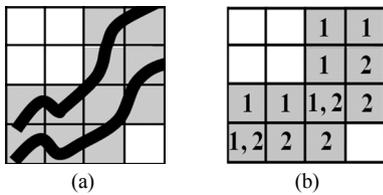


Fig. 9 Common partial searching

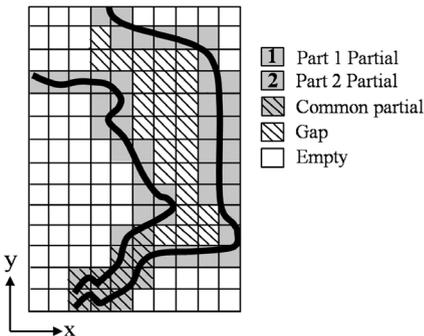


Fig. 10 Gap visualization

본 연구에서는 서로 다른 부품에 속하는 ‘부분’ 요소들이 서로 접하거나 하나의 ‘부분’ 요소에서 다수의 부품이 겹쳐지는 요소를 ‘공통 부분’(common partial) 요소라고 정의하고, 이 또한 틈새 영역으로 분류한다. 따라서 Fig. 8에서 찾아진 틈새 영역과 ‘공통 부분’ 요소를 추가함으로써 최종적으로 Fig. 10의 빗금친 요소와 같이 모든 틈새에 해당하는 영역이 자동으로 찾아진다.

5. 적용 예

본 연구에서 제안한 틈새 탐색 알고리즘의 효용성을 검증하기 위해, 자동차 차체 조립체에 대해 기존의 방법으로 틈새를 탐색하는 경우와 본 연구에서 개발한 프로그램으로 검증하는 방법을 비교해보고자 한다. 이를 위해 Fig. 11(a)와 같이 형상이 복잡하고 부품들이 삼차원 공간에서 조립되는 자동차 Outer panel 과 Front pillar 및 Inner pillar의 조립체의 경우를 비교하고자 한다. 기존의 방법에서는 Fig. 11(b)와 같이 이차원의 절단면을 생성하여 결합된 부품과 부품 사이의 틈새를 확인 하였다. 그러나 이러한 방법은 틈새가 발생하는 위치를 찾기가 어려우며, 절단면으로 틈새의 위치를 확인하더라도 전체적인 틈새의 형상을 시각적으로 확인하기가 어렵다.

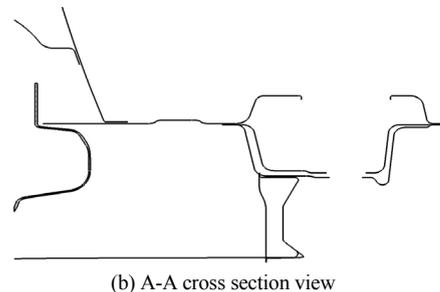
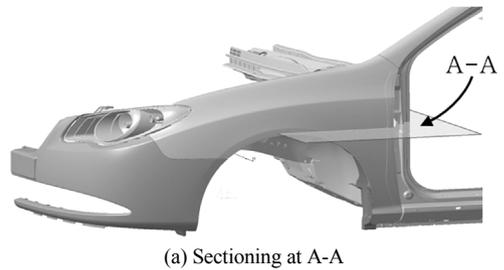
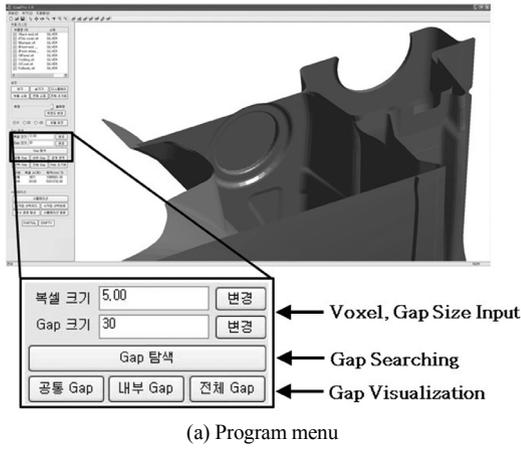
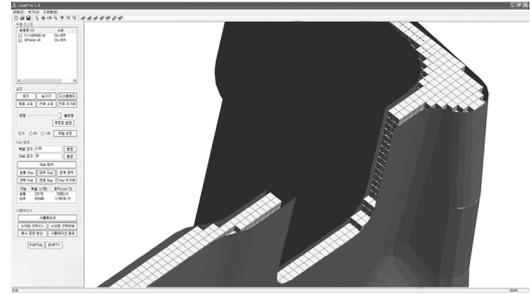


Fig. 11 Gap searching by conventional 2D method

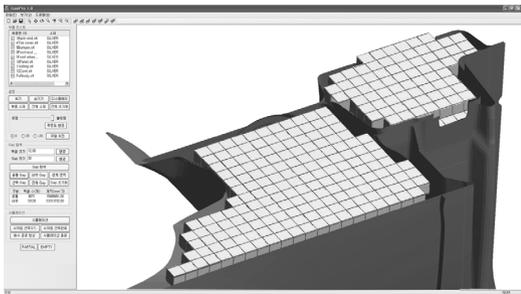


(a) Program menu

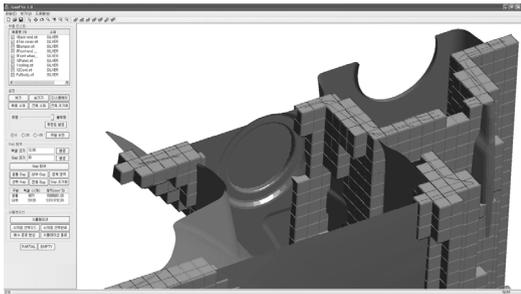


(e) Interior gap with fine elements

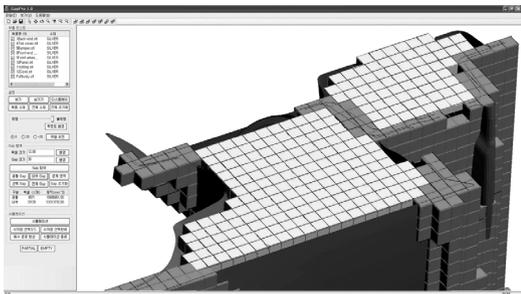
Fig. 12 Results of gap searching



(b) Interior gap



(c) Common partial gap



(d) Interior and common partial gap

한편 본 연구에서는 제안한 알고리즘을 Microsoft Visual Studio 2008의 환경에서 구현하여 Fig. 12(a)와 같은 틈새 탐색 프로그램을 개발하였다. Fig. 12(a)는 Fig. 11의 모델을 STL 파일로 읽어 들인 상태와 본 연구에서 개발한 프로그램 메뉴를 나타내는데, ‘Gap 탐색’ 버튼은 틈새를 탐색하는 기능이며, ‘공통 Gap’ 버튼은 ‘공통 부분’ 요소만을 가시화 하는 기능이며, ‘내부 Gap’ 버튼은 내부 ‘틈새’ 요소만 가시화 하는 것이며, ‘전체 Gap’ 버튼은 ‘공통 부분’ 요소와 ‘틈새’ 요소를 모두 나타내는 기능이다. Fig. 12(b)는 Fig. 11(b)의 형상에 대해 정확하게 부품들 사이의 내부 틈새 영역만을 탐색하고 가시화한 결과를 나타낸다. Fig. 12(c)는 부품들 사이의 틈새가 작아 내부의 틈새 영역으로 탐색되지 않았지만, ‘공통 부분’ 요소로 탐색된 결과를 나타내며, Fig. 12(d)는 내부의 틈새 영역과 공통부분 요소를 함께 나타낸 결과이다. 개발된 프로그램은 검사할 조립체의 분해 모델을 생성할 때, 해상도의 크기를 지정할 수 있는데, Fig. 12(e)는 동일한 모델에 대해 복셀의 크기를 줄여서 보다 상세하게 나타낸 결과이다. Fig. 12의 모델의 경우, 본 연구에서 개발한 프로그램으로 틈새를 탐색하는데 2초의 시간이 (조립 PC, Intel Core TM i7 CPU2. 94GHz, RAM 3.0 GB) 소요되었다. 따라서 본 연구에서 개발한 알고리즘과 프로그램을 이용하면 복잡한 형상의 조립체에 대해서도 설계자가 쉽고 빠르게 틈새 영역을 찾을 수 있다.

6. 결론 및 분석

본 연구에서는 옥트리와 복셀을 혼합한 방법을 이용하여 복잡한 형상의 조립체에서도 틈새를 자동

으로 탐색하고 가시화하는 알고리즘과 프로그램을 개발하였다. 항공기나 자동차와 같이 형상이 복잡하고 많은 수의 부품으로 이루어진 조립체를 설계할 때, 틈새를 확인하는 기존의 방법은 의심되는 곳의 이차원 단면도를 생성하여 확인하였다. 그러나 이러한 방법은 많은 수의 단면도를 확인해야 하기 때문에 시간이 많이 소요될 뿐만 아니라, 틈새의 위치를 누락할 수도 있고 틈새의 정확한 형상을 파악하기가 어렵다. 이에 비하여 본 연구에서 제안한 알고리즘을 이용할 경우 다음과 같은 장점이 있다.

- 1) 형상이 복잡한 부품들 사이의 틈새를 자동으로 탐색하고 틈새의 형상을 삼차원으로 가시화할 수 있다. 따라서 개발된 기능을 이용하면 설계자가 부품을 제작하거나 조립하기 전에 틈새를 쉽고 빠르게 확인할 수 있어 설계 오류를 줄일 수 있다.
- 2) 본 연구에서 채택한 옥트리와 복셀의 혼합 방법은 두 방법의 장점을 이용하기 때문에 복셀만을 이용하는 방법보다 훨씬 계산 시간이 빠르다. 또한 이를 이용하여 분해모델을 생성함으로써 부품의 복잡성에 제한을 받지 않고 틈새의 형상을 용이하게 확인할 수 있다.
- 3) 자유곡면으로 이루어진 부품들이 조립된 경우, 틈새와 외부 공간을 구분하기가 모호한 문제점이 있지만, 본 연구에서 제안한 탐색법은 이들의 경계면을 자동으로 찾을 수 있다.
- 4) 분해 모델로 형상을 나타내는 경우 요소의 크기보다 작은 틈새 영역을 찾지 못하지만, 본 연구에서는 ‘공통 부분’ 요소의 개념을 도입 하여 요소의 크기보다 작은 틈새 영역도 찾을 수 있다.
- 5) 본 연구는 자동차 차체의 부품을 대상으로 알고리즘과 프로그램을 개발하였지만, 개발된 기능은 비행기와 선박과 같은 대형 조립체에 적용 가능하다.

본 논문에서 제안한 방법은 앞에서 언급한 여러 가지 장점들을 가지고 있으나, 현재의 알고리즘은 PC 환경에서 승용차의 1/4 크기의 조립체를 탐색할 경우 2mm 이하의 미세 틈새를 정확하게 탐색하기에는 한계가 있다. 왜냐하면 본 연구에서 제안한 알고리즘은 내부적으로 복셀로 근사화하는데, 복셀의

크기가 작아질수록 소요되는 메모리와 이에 따른 탐색 시간이 급격히 증가하기 때문이다. 따라서 이러한 문제를 해결하기 위한 추후 연구 방향으로서 복셀을 절단하는 알고리즘을 이용하는 방법과, 조립체의 일부분만 국부적으로 선택하여 틈새를 탐색하는 연구가 필요할 것으로 사료된다.

후 기

이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.

References

- 1) S. H. Lee and D. S. Choi, "A Study on Concurrent Engineering Methodology for Automobile Wiring Harness Design System," Spring Conference Proceedings, KSAE, pp.607-618, 1993.
- 2) Y. H. Jung, "Implementation of Concurrent Engineering for Large Assembly Design: Part (I)," Transactions of the Society of CAD/CAM Engineers, Vol.2, No.2, pp.93-102, 1997.
- 3) I. H. Song and S. C. Chung, "Synthesis of the Digital Mock-up System for Heterogeneous CAD assembly," Computer in Industry, Vol.60, pp.285-295, 2009.
- 4) J. M. Park, I. H. Song and S. C. Chung, "Web-based Interference Verification System for Injection Mold Design," Transactions of KSME(A), Vol.30, No.7, pp.816-825, 2006.
- 5) L. Chen, Z. Song and L. Feng, "Internet-enabled Real Time Collaborative Assembly Modeling via an E-Assembly System : Status and Promise," Computer Aided Design, Vol.36, No.9, pp.835-847, 2004.
- 6) T. Chen, "Virtual Assembly of Mechanical Components and Collision Detection," Proceedings of International Conference on Electronics and Information Engineering, Vol.1, pp.443-447, 2010.
- 7) G. J. Jense, "Voxel-based Methods for CAD," Computer Aided Design, Vol.21, No.8, pp.528-533, 1989.

- 8) H. Streffen and M. Yavuz, "Extended Linked Voxel Structure for Point-to-mesh Distance Computation and Its Application to NC Collision Detection," *Computer-Aided Design*, Vol.41, pp.896-906, 2009.
- 9) A. Kela, "Hierarchical Octree Approximation for Boundary Representation-based Geometric Models," *Computer-Aided Design*, Vol.21, No.6, pp.355-362, 1989.
- 10) J. Cui, Y. Show and M. Zhang, "A Voxel-based Octree Construction Approach for Procedural Cave Generation," *Int. J. Computer Science and Network Security*, Vol.11, No.6, pp.160-168, 2001.
- 11) R. Aitor and A. Iker, "Automatic Extraction of the Topology of 3D Electrical Mock-ups Using a Mixed Octree-voxel Method," *Advances in Engineering Software*, Vol.40, pp.570-582, 2009.
- 12) J. W. Kim and S. H. Lee, "Fast Neighbor Cells Finding Method for Multiple Octree Representation," *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp.540-545, 2009.
- 13) H. Samet, "Neighbor Finding in Images Represented by Octree," *Computer Vision, Graphics and Image Processing*, Vol.46, pp.367-387, 1989.