

# 광조형물의 효율적 성형을 위한 지지대 자동 생성 시스템 개발

최 흥 태\*, 배 용 환\*\*, 이 석 희\*\*\*

## Development of Automatic Support Structure Generation System for Effective Building of Stereolithographic Parts

Hong-Tae Choi\*, Yong-Hwan Bae\*\*, Seok-Hee Lee\*\*\*

### ABSTRACT

Stereolithography is a process used to rapidly produce stereolithographic parts directly from three dimensional CAD models. However, design methodologies to create components to be built by stereolithography are different from those required by conventional machining processes. As a typical case in point, the support structures are required to support a component at build stage, but are later removed once building and curing are complete. These structures are used to anchor the component to the platform and to prevent sagging or distortion. This paper deals with the Support Map data structure prepared to find the appropriate regions of supports. Interferences between supports and parts, and support structure themselves as well, are checked and thus the system are searching the statically stable regions to remove the surplus support structures. Cross shaped tooth profiles are designed for easy eliminating the support structures.

**Key Words :** Stereolithography(광조형법), Support Structure(지지대), Support Map(지지대 맵)

### 1. 서 론

급속조형장치는 소프트웨어와 하드웨어를 포함하여 매우 고도로 일체화된 기기이다. CAD 데이터에서 2차원 슬라이스 데이터를 도입할 뿐만 아니라 실제의 조형에서는 그 데이터에 많은 손질을 해야 한다. 특히 광조형법일 경우 액체상태의 광경화성 수지를 사용하기 때문에 조형물이 완성되어서 수지통 속의 플랫폼으로부터 분리시킬 때 조형물의 파손 방지, 조형시 지붕형상과 돌출부분

에서의 처짐 방지, 그리고 섬과 같이 떠다니는 층의 고정 등을 위해서 지지대가 반드시 필요하다. 종래의 지지대 설계는 CAD상에서 조형대상을 모델링할 때 동시에 이루어지기 때문에 사용자의 주관적인 판단과 경험에 의해 시행착오를 거치면서 이루어져 왔다. 따라서 정확히 지지되어야 할 부분이 간과되거나 불필요한 부분에서 과잉 지지되는 경우가 발생하여 조형물의 최종 정밀도에 좋지 않은 영향을 미치고, 후 처리 공정이 길어지는 단점이 발생하였다. 이러한 문제점을 해결하기 위한 많은 노력으로 현

\* 경남정보대학 차량기계과  
 \*\* 안동대학교 기계공학교육과  
 \*\*\* 부산대학교 기계공학부 및 기계기술연구소

재 BridgeWorks<sup>(1)</sup>, Magics SG<sup>(2)</sup>와 같은 지지대 설계를 위한 상용 소프트웨어가 시판되고 있다. 그러나 이러한 상용 소프트웨어는 개발비 및 상업적인 이유로 인해 상당히 고가로 시판되고 있으며 그 내부 알고리즘은 밝혀진 바가 없다.

본 논문에서는 신속한 검색을 위한 데이터 구조와 효율적인 알고리즘에 의해 조형 모델을 조형방향에 대해 최소한의 사용자 입력으로 지지대 데이터를 자동으로 생성하고 STL 포맷으로 출력하는 방법을 제시하고자 한다.

## 2. 관련연구

일본의 吉川浩一<sup>(3)</sup> 등은 조형대상을 한 층씩 적층할 때 컵의 손잡이 아랫부분에서와 같이 광경화성수지 위에 떠 있는 섬과 같은 층을 지지하기 위해 로봇트 동작계획법중 Backprojection법을 응용하여 조형물과 간섭이 일어나지 않는 접지가능원추를 구하고 지지할 곳에서부터 플랫폼 포옴 방향으로 지지대를 수직으로 내릴 때 접지가능원추와 만날 경우 간섭이 일어나지 않는 접지가능영역으로 경로를 바꾸었다. 그러나 조형물이 복잡할 경우 접지가능원추를 구하기가 어려운 문제점이 있다.

Kirschman<sup>(4)</sup> 등은 facet 모델을 이루는 모서리들을 Base border, False border, Flat border, Beam border로 분류하고, 이러한 facet의 경계 모서리를 검색할 수 있도록 트리 구조의 자료구조를 이용하였다. 그러나 이러한 검색법만으로는 조형 대상과 지지대와의 간섭 검사를 할 수 없으며, 단순 지지보와 같은 형상에서는 과도한 지지대가 생길 수가 있다.

Webb<sup>(5)</sup>은 Stereolithography를 위한 STL파일로부터 지지대를 생성하는 MS Windows 프로그램인 Support Designer의 개발에 대해 발표하였지만 세부 알고리즘에 대한 설명이 없고, 직육면체에 구멍이 있는 간단한 형상의 적용에만 소개하고 있다.

Swaelens<sup>(6)</sup> 등은 벨기에 Materialise NV의 지지대 자동생성 프로그램인 MAGICS 소프트웨어에 관한 내용으로 STL 파일에서 지지대가 세워질 영역의 돌출부분, 길이, 면적 등을 고려하여 Block, Line, Point, Gusset 지지대중 하나를 결정하는 알고리즘을 제시하였다. 그러나 이러한 알고리즘으로는 비교적 단순한 다면체 형상에만 적용 가능하며 지지할 부분이 경사면일 경우와 조형대상과의 간섭이 있을 경우에는 정확한 지지대를 생성할 수 없다.

허정훈<sup>(7)</sup> 등은 SLA를 이용한 시작작업의 효율화 방안으로 최적의 성형방향을 찾기 위한 알고리즘을 제시하였는데 성형 정확도 향상을 위해, 계단효과에 의해 생성된 계단부분의 면적을 최소화하는 목적함수를 무차원화하였으며 이를 위해서 가변 층 두께 개념을 도입하였다. 그러나 최적 성형 방향을 결정하기 위해서 모든 facet의 법선 벡터를 z축에 일치시켜가며 목적함수를 계산하기 때문에 검색 및 계산시간 측면에서 바람직하지 못하다.

## 3. 본 론

광조형법에서 지지대는 큰 단점 중의 하나이지만, 액체 광경화성 수지에 레이저를 주사하여 한 층씩 적층 하는 특성 때문에 지지대는 반드시 필요하다. 초기에 주로 사용된 아크릴레이트(acrylate) 수지는 내부응력 때문에 적층시 경계부분에서 말리는 현상(curling)이 발생한다. 이를 최소화하기 위해서 지지대가 필요하며, 말리는 힘(curling force)을 견딜 만큼 지지대를 강하게 만들다 보니 후 처리 공정이 길어지는 문제점이 발생하였다. 이에 비해 최근에 개발된 에폭시(epoxy)와 비닐에테르(vinyl-ether) 수지를 사용할 경우 말리는 정도가 적기 때문에 주로 레이저 주사중 조형 대상의 슬라이스가 이 전에 없던 섬(island)과 같은 단면이 발생할 때 이 부분을 지지할 지지대가 필요하다.

접성이 큰 수지 속에서 엘리베이터의 상하운동 때문에 수지 표면영역이 파도가 일어나는 현상을 'Parachuting'이라 하고, 위층의 하층으로 인해 처짐이 발생하는 현상을 'Sagging'라 하며, 수지가 경화되면서 수축에 의한 뒤틀림이 발생하는 현상을 'Warping'이라고 한다. 이상은 초기 광조형법의 대표적인 문제점으로 지적된 사항들이지만, 최근의 상용 광조형 장치 개발업체에서 조형성이 뛰어난 광경화성 수지 개발 및 수지 공급장치 개선 등으로 조형물의 정밀도를 많이 향상시키고 있다. 그러나 여전히 지지대는 소프트웨어적으로 처리해야 할 부분으로 조형의 정확성과 효율성 측면에서 고려하여야 할 사항이 많다.

### 3.1 지지대의 필요성

Fig. 1에서는 지지대가 필요한 조형 대상의 자세와 지지 형태를 5가지로 분류하였다.

(a), (b)는 가장 일반적인 경우로 삼각형 facet의 법선 벡터의 z 값이 음의 값(-1.0~-0.87)을 가지는 경사면

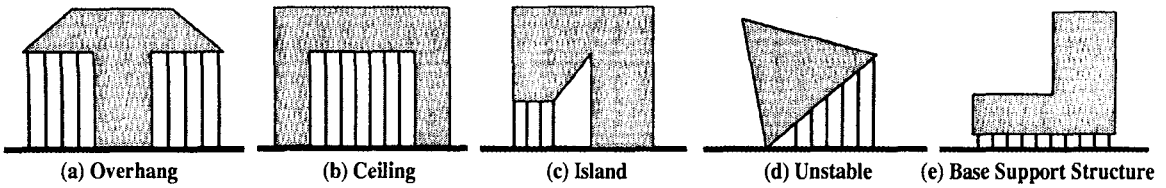


Fig. 1 Various Support Types

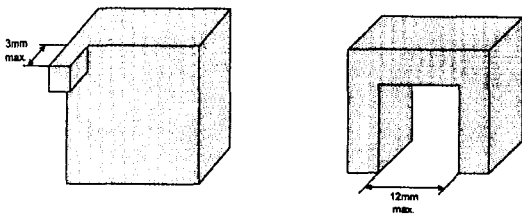


Fig. 2 Maximum unsupported length

중에서 일정한 각도( $0^\circ \sim 30^\circ$ )이내에 드는 수평방향으로 돌출된 부분으로 처짐이 발생한다. (c)는 레이저 주사중 조형 대상의 슬라이스가 이 전의 층에 없던 섬(island)과 같은 단면이 발생하는 경우로 지지대가 없으면 레이저 주사와 동시에 이 부분이 바닥 면에 가라앉는다. (d)는 설계자가 지지대를 고려하지 않고 3차원 CAD에서 조형 대상을 모델링할 때 흔히 발생한다. 이러한 경우 (a), (b)의 경우에서와 같이 지지대가 필요하지만, STL 파일을 대상으로 조형모델의 최적자세를 구하는 전처리 공정에 의해서 수정할 수 있다. (e)는 조형물이 완성된 후에 조형물 내부에 고립되어 있는 미경화 수지 제거, 표면에 묻어 있는 수지 세척, UV 오븐에서의 건조, 지지대 제거 등과 같은 후처리 공정을 위해 플랫폼용 바닥 면으로부터 조형물을 분리할 때 파손과 변형을 방지하기 위해 Base Support가 필요하다.

Fig. 2에서는 Fig. 1의 (a), (b)에 해당하지만 스스로 지지할 수 있는 최대 길이를 도식적으로 보여주고 있다. 일반적으로 외팔보와 같이 돌출된 부분이 3mm이하이면 스스로 지지될 수 있으며, 단순 지지보 형태는 그 사이의 간격이 최대 12mm까지는 지지대 없이 견딜 수 있기 때문에 이러한 경우 지지대가 필요하지 않다<sup>(8)</sup>.

### 3.2 STL 포맷의 위상정보 재구성

Fig. 3에서와 같이 사면체를 구성하는 삼각형 facet수와 vertex수는 각각 4개이다. 그러나 이러한 사면체를

STL 포맷으로 나타내면 삼각형 facet수는 4개로 같지만, vertex수는 12개로 훨씬 많다. 이는 Vertex-to-Vertex 규칙을 따르는 STL 포맷의 특성 때문에 vertex 수가 중복되게 존재하게 된다. 주어진 STL파일로부터 중복되지 않는 총 vertex수와 edge수는 식(1), (2)와 같이 구할 수 있다.

$$\text{중복되지 않는 총 vertex수} = \frac{\text{총facet수}}{2} + 2 \quad (1)$$

$$\text{중복되지 않는 총 edge수} = \text{중복되지 않는 총 vertex수} \times 3 - 6 \quad (2)$$

이러한 중복된 vertex 수를 줄이기 위해서 Vertex와 Facet 구조체 리스트를 작성하여 새로운 위상정보를 구성한다. Fig. 3에서 각 vertex를 공유하는 삼각형 facet 집합을 다음과 같이 구성할 수 있다. 이는 Vertex 구조체 리스트를 구성하는 기본 개념으로 검색 시간을 줄이기 위함이다.

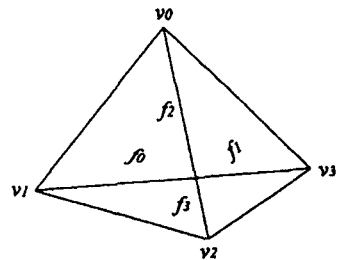


Fig. 3 Removed redundancy vertices

$$\begin{aligned} v0 &= \{f0, f1, f2\} \\ v1 &= \{f0, f2, f3\} \\ v2 &= \{f0, f1, f3\} \\ v3 &= \{f1, f2, f3\} \end{aligned}$$

Facet 구조체 리스트의 요소로써 지지할 영역과 이웃한 미지지 영역과의 정역학적 관계를 검색하는데 도움을

주기 위해서 Fig. 3의 각 삼각형 facet의 세 이웃 facet 집합을 다음과 같이 구성할 수 있다.

$$\begin{aligned}
 f_0 &= \{f_1, f_2, f_3\} \\
 f_1 &= \{f_0, f_2, f_3\} \\
 f_2 &= \{f_0, f_1, f_3\} \\
 f_3 &= \{f_0, f_1, f_2\}
 \end{aligned}$$

### 3.3 Vertex와 Facet 구조체 리스트

위상 정보가 부족한 STL 포맷으로부터 지지대를 설계하기 위해서는 무엇보다 효율적인 데이터 구조가 중요하다. 따라서 본 연구에서는 지지대가 요구되는 영역, 지지대와 조형 대상과의 간섭 검사 및 과도한 지지대 방지를 위한 검색 자료로써 Vertex 리스트와 Facet 리스트를 구성하여 사용한다. Vertex 리스트는 정점의 x, y, z좌표 값과 이를 공유하는 삼각형 facet의 번호를 가진다.

Facet 리스트는 삼각형 facet의 법선 벡터, 세 정점의 vertex 번호, 세 이웃 삼각형 facet 번호, 지지대 필요 여부 등의 정보들을 가진다. 이러한 위상 정보들은 Vertex

리스트와 Facet 리스트를 동시에 병행해서 구성되어진다. Fig. 4에서는 Vertex 리스트와 Facet 리스트 구성 절차를 번호 순서대로 도식적으로 보여 주고 있으며 세부적인 설명은 다음과 같다.

① 입력 STL 파일로부터 한 삼각형 facet 단위로 vertex 좌표 값을 읽어 Vertex 리스트에 등록한다. 만일 동일한 점이 이미 등록되어 있다면 이 vertex의 색인 번호 (StlVertex(i)의 i)를 리턴하고, 동일한 점이 등록되어 있지 않으면 새로운 vertex 색인 번호를 추가하고 i를 리턴 한다.

② 입력 STL 파일로부터 읽어들이는 법선 벡터의 x, y, z 좌표 값을 Facet 리스트 배열인 StlFacet(i)의 Normal.x, Normal.y, Normal.z에 할당한다. 그리고 Normal.z값이 -1.0~-0.87 사이에 있으면 지지대가 필요한 것으로 판별하여 NeedSupport에 'YES' 플래그를 설정하고, 그렇지 않으면 'NO' 플래그를 설정한다.

③ ①에서 리턴된 세 StlVertex(i)의 vertex 색인 번호 i를 할당한다.

④ StlFacet(i)의 facet 번호 i를 StlVertex(i)에 연결

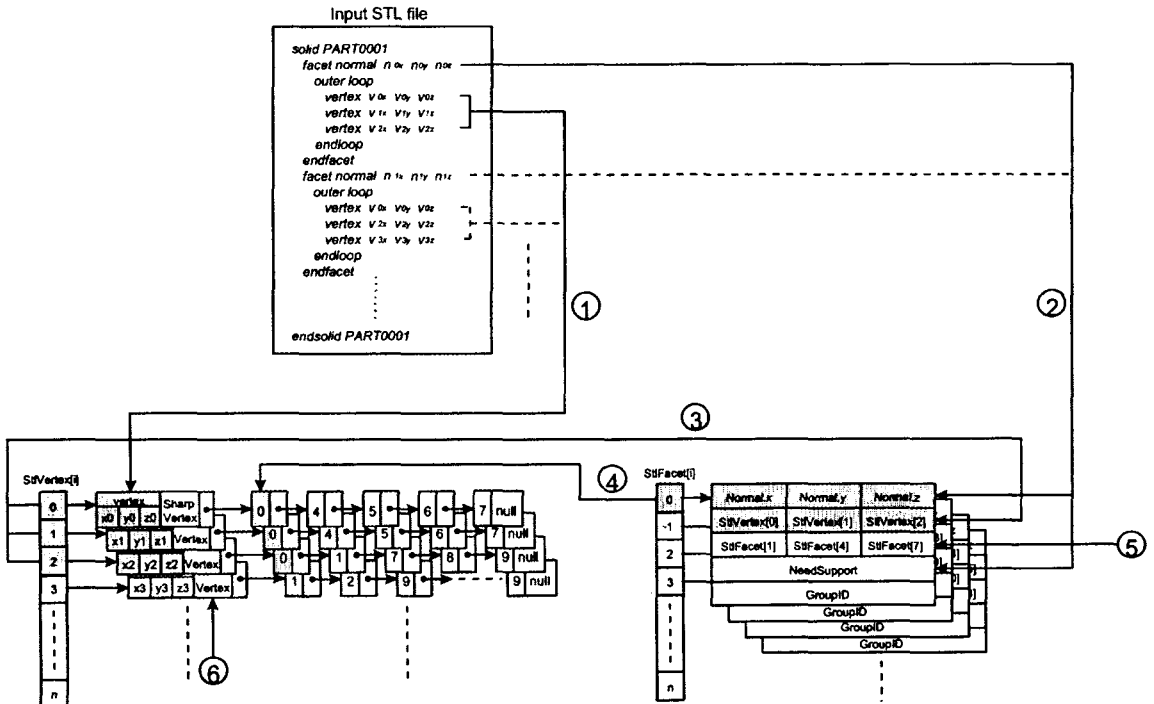


Fig. 4 Construction of vertex and facet list

시킨다.

이상과 같은 단계를 입력 STL 파일의 삼각형 facet 개수만큼 반복해서 수행함으로써 일차적인 위상정보를 구성하게 된다. ⑤, ⑥은 각 삼각형 facet의 세 이웃들을 찾아내고, 아래쪽으로 향한 침점 여부를 판별하는 단계이다. 3.4에서 검색방법을 설명한다.

### 3.4 이웃 삼각형 facet 및 아래 방향 침점 검색

3.3에서 중복되지 않는 vertex와 facet들의 위상정보들이 완성되면 이를 바탕으로 삼각형 facet당 세 이웃 facet들을 찾는 문제는 다음과 같은 방법으로 쉽게 구할 수 있다.

Fig. 5에서 삼각형 facet  $f_0$ 의 세 이웃을 찾기 위해서 먼저, Facet 구조체 리스트에서  $f_0$ 를 이루고 있는 세 정점의 vertex 번호(StlVertex[i]의 i)를 찾아서 해당 vertex를 공유하는 facet 번호(StlFacet[i]의 i)를 Vertex 구조체 리스트에서 검색한 결과는 다음과 같다. 여기서 이해를 돕기 위해서 StlVertex[0]는  $v_0$ 로, StlFacet[0]는  $f_0$ 로 표기한다.

$$\begin{aligned} v_0 &= \{ f_0, f_1, f_2, f_3, f_4 \} \\ v_1 &= \{ f_0, f_4, f_5, f_6, f_7 \} \\ v_2 &= \{ f_0, f_1, f_7, f_8, f_9 \} \end{aligned}$$

위 세 정점  $v_0, v_1, v_2$  집합의 원소들은 삼각형 facet  $f_0$ 의 이웃이 될 후보 facet들이다.

이를 기초로 하여 삼각형 facet  $f_0$ 의 세 모서리  $e_0, e_1, e_2$ 를 공유하는 facet들은 다음과 같이 각 모서리를 이루는 정점의 교집합으로 구할 수 있다.

$$\begin{aligned} e_0 &= v_0 \cap v_1 = \{ f_0, f_4 \} \\ e_1 &= v_1 \cap v_2 = \{ f_0, f_7 \} \\ e_2 &= v_2 \cap v_0 = \{ f_0, f_1 \} \end{aligned}$$

위 세 모서리  $e_0, e_1, e_2$  집합의 원소들은 삼각형 facet  $f_0$  자신을 포함한 이웃 facet들을 가리킨다. 따라서 자기 자신을 제외한 세 이웃 facet들은 다음과 같이 구할 수 있다.

$$\begin{aligned} f_0 \text{의 세 이웃 facet 집합} &= \{ e_0 \cup e_1 \cup e_2 \} - \{ v_0 \cap v_1 \cap v_2 \} \\ &= \{ f_0, f_1, f_4, f_7 \} - \{ f_0 \} \\ &= \{ f_1, f_4, f_7 \} \end{aligned}$$

Fig. 6에서 아래쪽으로 향한 침점은 경사면의 기울어진 각도만으로 판별하면 지지대가 요구되지 않는 면이지만, 지지대가 없으면 조형되지 않는 아주 민감한 부분이다. 이러한 뾰족한 정점을 공유하는 모든 facet의 법선 벡터 z값은 지지대가 요구되지 않는 음의 값(-0.86~-0.01)을 가진다. 이 부분에 대한 지지대의 설계를 위해서는 Vertex 구조체 리스트(StlVertex[i])에 연결되어 있는 facet 번호에 해당하는 Facet 구조체 리스트(StlFacet[i])를 찾아서 facet의 법선 벡터 z값을 검색하여 일정한 범위내의 음의 값을 가지면 Sharp Vertex로 판별하고, Vertex 구조체 리스트의 Sharp Vertex flag를 'ON' 시킴으로써 Support Map 생성 시에 반영한다.

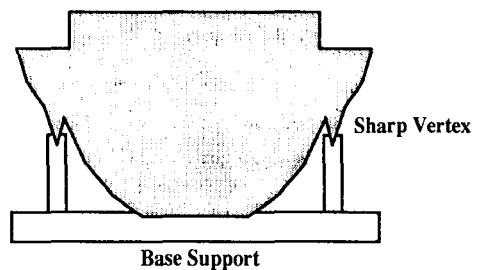


Fig. 6 Sharp Vertex Support

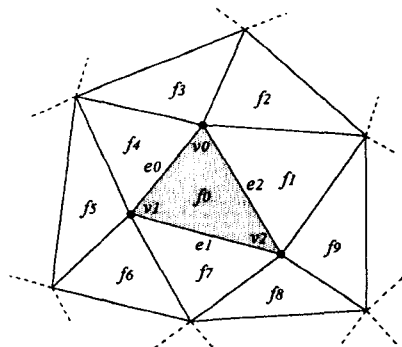


Fig. 5 Searching of three neighboring facets

### 3.5 지지대와 조형물 사이의 오프셋

Fig. 7은 수평으로 돌출된 외팔보와 같은 형상이고, Fig. 8은 단순 지지보와 같은 형상으로 지지대가 필요한 영역은 삼각형 facet  $F_1, F_2$  부분이다. 그러나 이 부분 전체를 바닥 면에서 조형 방향으로 지지대를 만들게 되면 조형 대상의 벽과 지지대가 붙게되어 후처리 공정에서 지지대를 분리하는 작업이 어렵게될 뿐만 아니라 조형물의 표면정도에 좋지 않은 영향을 미친다. 따라서 본 연구에

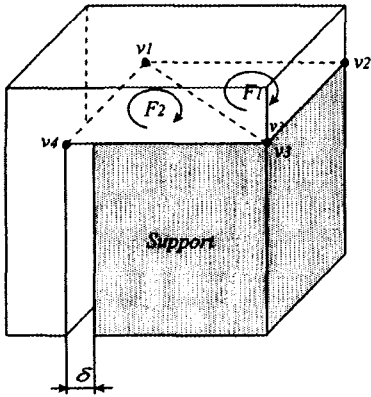


Fig. 7 Support structure of cantilever with offset

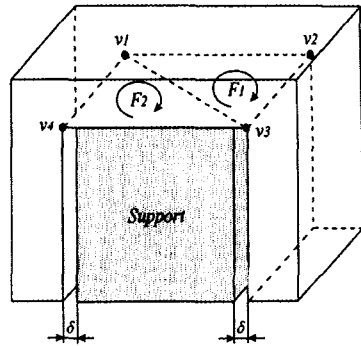


Fig. 8 Support structure of simple beam with offset

서는 Support Map에서  $\delta$ 만큼 오프셋된 지지대를 설계한다. 여기서  $\delta$ 값은 사용자 입력으로 받아들이며 내장 값은 6mm로 설정하였다.

#### 4. 지지대 결정 알고리즘

##### 4.1 동일한 지지 영역의 삼각형 facet 그룹화

기본적인 자료구조가 완성이 되면 Support Map상에서 지지대가 필요한 영역중 정역학적으로 안정된 부분의 지지대를 제거하기 위해 지지대가 필요한 삼각형 facet별로 그룹화 시킨다.

Facet 구조체 리스트의 NeedSupport 멤버 값이 'YES'이고 GroupID가 'NONE'인 facet을 찾으면 같은 GroupID를 가질 삼각형 facet을 검색하기 시작한다. 여기서 GroupID는 'NONE'으로 초기화 되어있다. Fig. 9에서는 각 삼각형 facet의 세 이웃 facet들을 검색하기 위해 재귀 호출하는 상태를 나무구조로 보여주고 있다. 이들 이웃한 삼각형 facet들을 검색하고 NeedSupport와 GroupID를 검사하여 GroupID를 결정한다.

##### 4.2 Support Map 작성

Vertex와 Facet 구조체 리스트가 완성되면 실제 지지

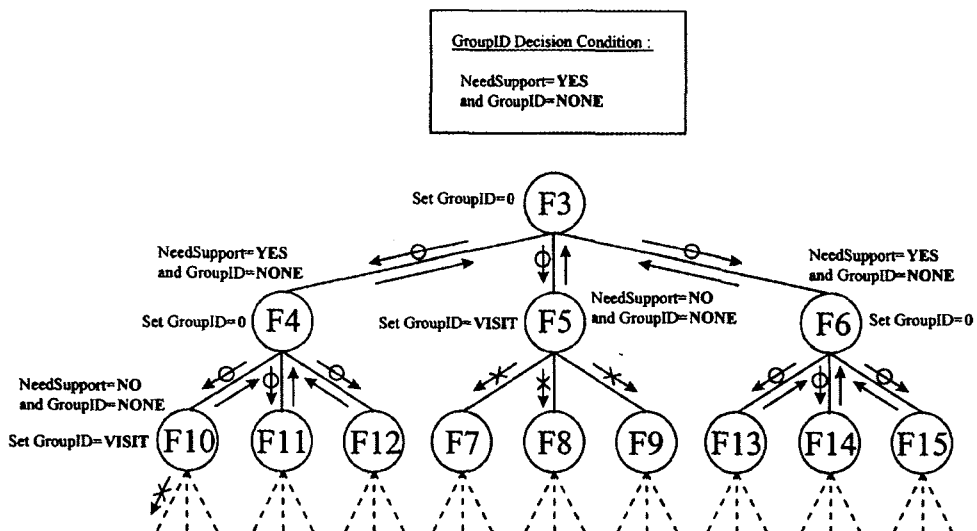


Fig. 9 Tree structure for GroupID decision

대 생성에 필요한 데이터를 작성하기 위해 Fig. 10과 같은 Support Map이란 개념을 도입하였다. 여기서 Support Map은 바둑판 모양의 격자로 이루어져 있으며 격자 점마다 SupportMap(i, j)와 같은 자료구조를 가진다. Support Map의 크기는 플랫폼 바닥 면의 넓이와 같기 때문에 최대 조형범위와 일치한다.

Support Map 구조체 리스트 작성을 위해 먼저 삼각형 facet  $f1 \sim f8$ 을 순서대로 Support Map상에 투영을 시킨다. 이 때 만나는 격자 점에 해당하는 Support Map (i, j)에 삼각형 facet 번호, 지지대 유·무, Support Map 바닥 면에서 투영된 삼각형 facet까지의 높이, 지지대 바닥 면과 꼭대기 z좌표값을 구하여 할당한다.

Fig. 11의 (a)에서는 삼각형 facet  $f8, f1, f5, f4$ 를

Support Map상에 투영시킬 때 SupportMap(6, 3)에서 만나는 예를 나타낸 것이다. Fig. 11의 4개의 삼각형 facet을 투영할 때 SupportMap(6, 3)에서 4개의 구조체가 포인터 변수 Next에 의해 리스트로 연결되는 과정을 순서대로 보여주고 있다.

여기서 GroupID는 같은 영역을 지지할 삼각형 facet 끼리 묶기 위해서 부여하는 번호이다. Fig. 10의 예에서는 삼각형 facet  $f5, f6$ 이 같은 그룹으로 GroupID가 '0'으로 설정되고, 삼각형 facet  $f7, f8$ 은 GroupID가 '1'로 설정된다.

#### 4.3 불필요한 지지대 제거

Fig. 12에서 삼각형 facet  $f7, f8$ 은 지지대가 필요한 영

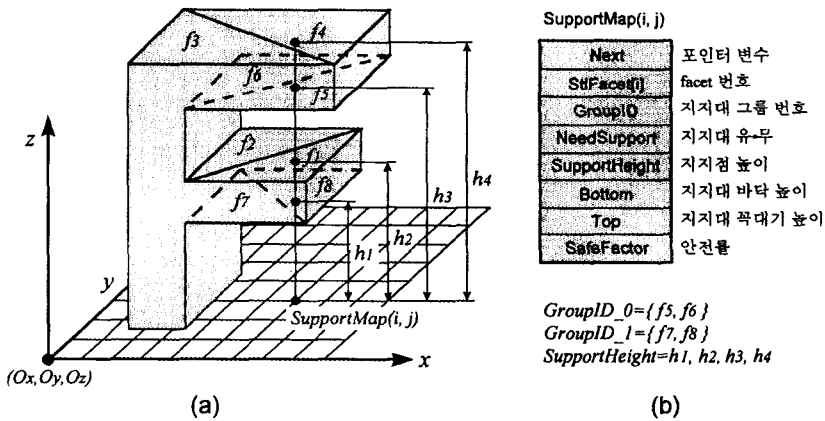


Fig. 10 Structure List of Support Map

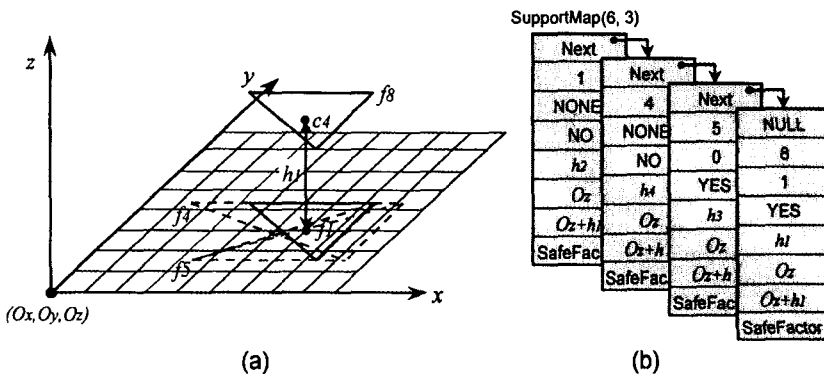


Fig. 11 Writing process of Support Map structure list

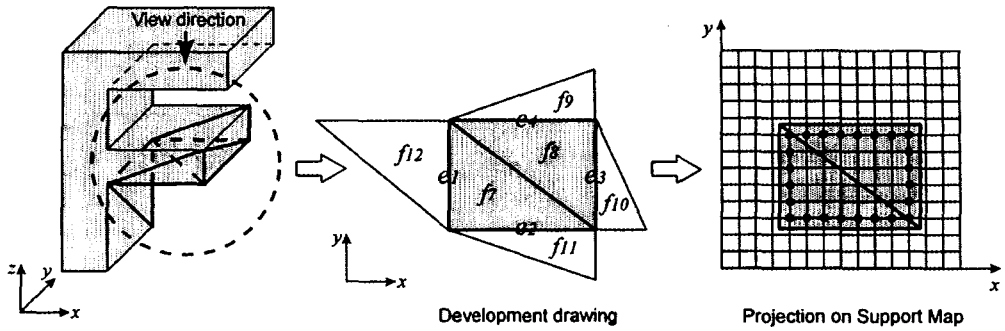


Fig. 12 Searching of removing Support Map

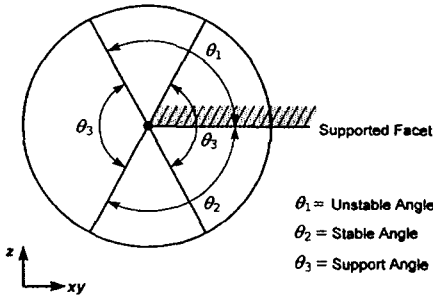


Fig. 13 Static status decision circle of two between facets

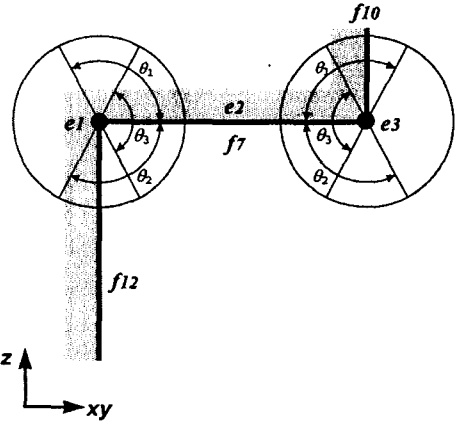


Fig. 14 Application of stable or unstable decision circle

역으로 GroupID가 '1'로 설정되어 있다. 따라서  $f_7, f_8$ 과 접해 있는 facet  $f_9, f_{10}, f_{11}, f_{12}$ 를 검사하여 GroupID가 '1'인 삼각형 facet  $f_7, f_8$ 과의 정역학적 관계를 조사한 후 안정 영역 또는 불안정 영역으로 판별하여 해당 SupportMap(i, j)상의 SafeFactor에 제거해야할 길이값을 할당한다. 모서리  $e_1$ 에서  $f_7$ 과  $f_{12}$ 는 정역학적으로 안정한 영역으로 판별하여 SafeFactor에 스스로 지지할 수 있는 단순 지지보의 길이의 반인 6mm를 할당한다. 나머지  $e_2 \sim e_4$ 는 불안정한 영역으로 판별하여 0mm를 할당한다.

Fig. 13에서는 GroupID가 같은 삼각형 facet들, 다시 말해서 동일한 영역을 지지해야할 삼각형 facet들과 이들의 경계에 인접해 있는 facet들이 서로 안정한지, 불안정한지에 대한 정역학적 관계를 판별하기 위해 사용되는 원을 나타내고 있다. 여기서,  $\theta_3$ 는 지지대가 요구되는 삼각형 facet(-1.0 ≤ 법선벡터 z값 ≤ -0.87)이 위치하는 각도이고,  $\theta_1$ 과  $\theta_2$ 는 이와 인접한 삼각형 facet이 위치하는 각도를 나타내며 각각 불안정과 안정한 영역을 판별하게 된다.

Fig. 14는 Fig. 12의 점선 원에 대한 예로써 지지대가 필요한 삼각형 facet은  $f_7$ 이며 이웃한 삼각형 facet은  $f_{10}$ 과  $f_{12}$ 이다. 여기서  $f_7$ 이 존재할 수 있는 범위는  $\theta_3$  이내이고, 이웃한 삼각형 facet  $f_{10}$ 과  $f_{12}$ 는  $\theta_1$  또는  $\theta_2$  이내에 존재하게 된다. 따라서  $f_{10}$ 은  $\theta_1$ 에 위치하기 때문에 수평방향으로 돌출된 형상으로 모서리  $e_3$  쪽은 불안정하다.  $f_{12}$ 는  $\theta_2$ 에 위치하기 때문에  $f_7$ 에 대해서 벽과 같은 상태로 지지하므로 모서리  $e_1$  쪽은 안정하다고 판별한다.

#### 4.4 Tooth Profile 및 지지대 데이터 출력

이상과 같이 Support Map 데이터가 완성되면 지지대를 STL 포맷으로 출력하게 된다. 조형물의 지지면과 지지대 사이가 단순히 접해있는 것만으로는 조형물을 확실히 지지할 수 없기 때문에 Fig. 15에서와 같이  $\delta_2$ 만큼 지지대의 일부가 조형물의 내부에 박혀 있어야 한다. 본 시



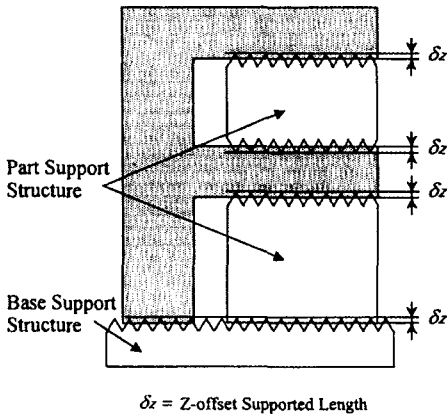


Fig. 15 Z-offset supported length

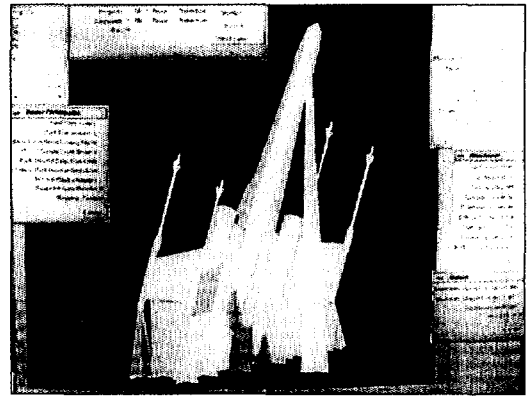


Fig. 17 Support structure of X-Wing (Maestro)

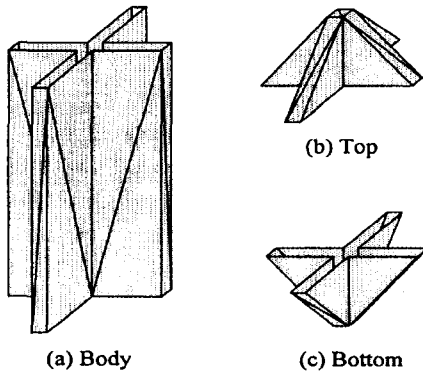


Fig. 16 Component of support structure

시스템에서는  $\delta z$ 값을 사용자가 입력하도록 하였지만 내정값으로 1mm를 가진다. 그러나 Fig. 16의 (a)와 같이 지지대 몸체 부분이 지지면 안쪽으로 박혀있을 경우 조형후 지지대 분리가 어렵게 된다. 그래서 Fig. 16의 (b), (c)와 같은 십자형 모양의 Tooth Profile이 지지대의 몸체와 지지면을 연결함으로써 후 처리 공정을 단순화시킬 수 있으며 정확히 지지할 수 있다.

### 5. 적용예

본 시스템에서 적용한 예는 X-Wing으로 비교적 형상이 복잡하고 날개 사이의 영역과 아래로 향하는 첨점부분에 대한 지지대 생성 여부를 알아볼 수 있다. Fig. 17은 3D System사의 SLA 지원 소프트웨어인 Maestro에서

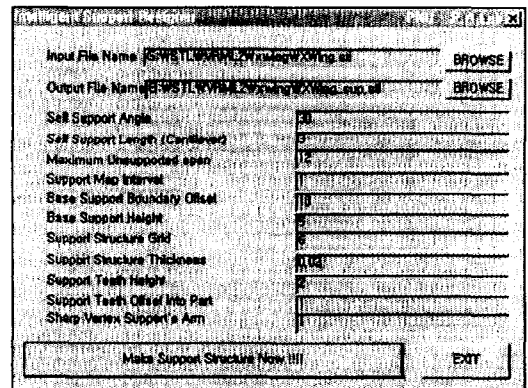


Fig. 18 Support generation parameter for X-Wing geometry

생성된 지지대를 보여주고 있으며 Fig. 19~Fig. 22는 Visual C++4.1과 Open Inventor를 사용하여 개발된 IntelliSL에서 자동 생성된 지지대를 나타내고 있다.

Fig. 18은 IntelliSL의 지지대 생성 조건 입력창으로 사용자 입력에 따라 X-Wing 모델을 0°, 30°, 70°, 90° 자세로 변화시켜 가며 적용한 결과 Fig. 19~Fig. 22와 같은 지지대가 생성됨을 보여주고 있으며, 생성 결과는 Table 1에 나타내었다. Fig. 23은 에폭시 수지를 사용하여 광조형장치에서 제작된 X-Wing 프로토타입이다.

조형물의 경사가 0°와 30°인 경우에 'X'자 날개 사이에 Part support가 간섭 없이 생성되었으며, 날개를 형성하는 모든 facet이 XY평면으로부터 60° 이상 기울어져 있는 70°와 90°에서는 support가 생성되지 않음을 확인할 수 있었다.

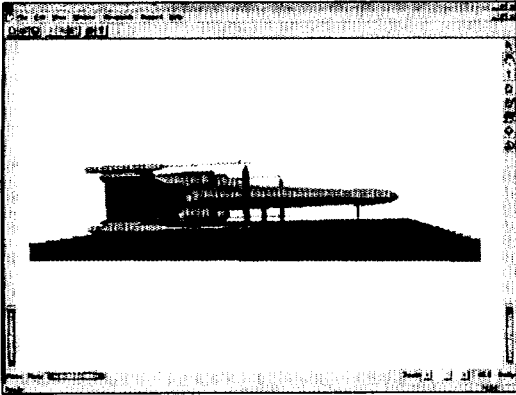


Fig. 19 Horizontal part

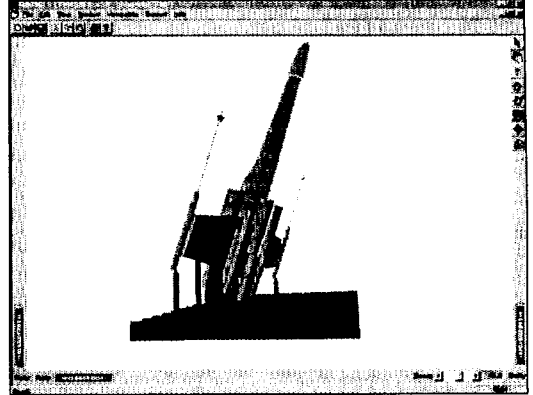


Fig. 21 Part inclined by 70°

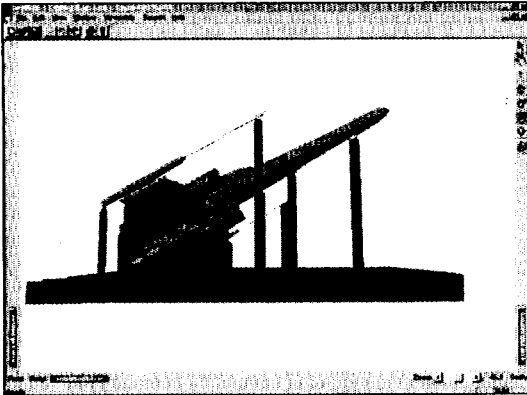


Fig. 20 Part inclined by 30°

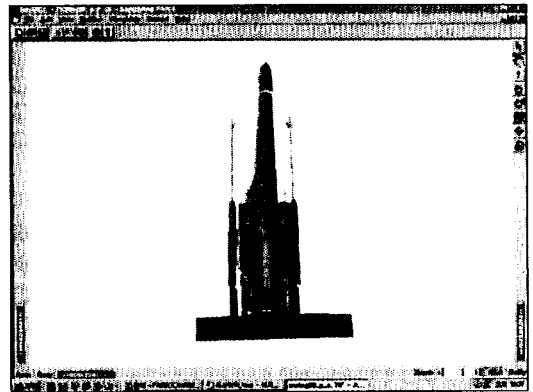


Fig. 22 Vertical part

Maestro와 IntelliSL의 지지대 자동 생성 결과인 Fig. 17과 Fig. 21을 비교하면 아래와 같다.

(1) 두 시스템 모두 지지가 필요한 영역과 Downward sharp vertex에 대해 지지대를 생성한다.

(2) Maestro는 경사 각도로 볼 때 지지가 필요한 영역을 모두 지지하였으나 IntelliSL은 경사 각도상 지지가 필요하더라도 이웃 facet과의 관계를 고려하여 정역학적으로 안정한 영역에 대해서는 지지대를 생성하지 않는다.

(3) 두 시스템 모두 조형물 사이에 지지대를 생성할 경우 지지대 상·하부에 tooth profile을 생성하여 수지가 쉽게 유출되도록 하였다. Maestro에서는 생성된 지지대 끼리 간섭을 일으키나 IntelliSL은 각 팔의 길이가 가변적인 십자형 지지대를 사용하기 때문에 지지대와 조형물이 간섭을 일으키지 않는다.

Table 1 Support generation of differently sloped part

Geometry		Fig. 19	Fig. 20	Fig. 21	Fig. 22
P a r t	Slope from x-y plane	0°	30°	70°	90°
	Area projected on x-y plane	70×100mm	70×91.5mm	70×44.3mm	70×20mm
	Height	20mm	56mm	96mm	100mm
	Number of vertex	18,696			
	Number of facet	6,232			
STL file size		1.2MByte			
S u p p o r t	Number of vertex	48,318	51,648	23,889	13,884
	Number of facet	16,106	17,216	7,963	4,628
	STL file size	4.2MByte	4.5MByte	2.1MByte	1.2MByte
	Number of part support	17	25	12	16
	Number of between-part support	16	14	0	0
r t	Number of downward vertex support	8	15	9	0
	Processing time	25 sec	26 sec	22 sec	18 sec
CPU		PENTIUM® processor, 200MHz			

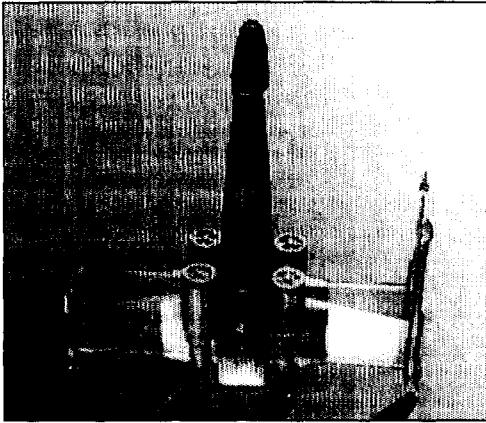


Fig. 23 Built X-Wing prototype

## 6. 결 론

본 연구에서는 광조형물의 효율적인 성형을 위한 지지대 자동생성 시스템인 IntelliSL을 개발하였다. 지지대가 필요한 영역에 정확히 지지대를 생성시키기 위해 독자적인 Support Map을 구성하여 조형물과 지지대 사이의 간섭, 지지대끼리의 간섭, 조형물의 기하학적 조건을 고려하여 불필요한 영역에서 지지대 생성을 방지하도록 하였다. 그리고 지지대와 조형물의 접촉부분에 Tooth profile를 사용함으로써 조형물로부터 지지대의 제거를 용이하도록 하였다. 생성된 지지대 데이터는 이후의 슬라이싱 공정에 일관되게 적용시키기 위해 STL 파일 포맷으로 출력하였다. 개발된 IntelliSL을 여러 형상에 적용한 결과 지지대가 정확히 생성됨을 알 수 있었으며 이를 조형물 성형에 이용하여 원하는 프로토타입을 얻을 수 있었다.

## 후 기

본 연구는 97년도 과학재단 핵심전문연구과제비에 의해 수행되었으며, 이에 관계자 제위께 감사드립니다.

## 참 고 문 헌

1. BridgeWorks, Solid Concepts Inc., <http://solid-concepts.com/bridgewk.html>.
2. Magics SG(Support Generation), Materialise,

<http://www.materialise.com/technic/magsg.html>.

3. 吉川浩一, 鈴木宏正, 木村文彦, “光造形法におけるサポート形状成形法,” 日本精密工學會春季大會學術講演會講演論文集 C61, pp. 819~820, 1995.
4. C. F. Kirschman, C. C. Jara-Almonte, A. Bagchi, R.L. Dooley, A.A. Ogale, “Computer Aided Design of Support Structures for Stereolithographic Components,” Proceedings of the 1991 ASME Computers in Engineering Conference, Santa Clara, CA, pp. 443~448, 1991.
5. D. Webb, V. Gerdes, C. Chassapis, “Computer-Aided Support-Structure Design for Stereolithography Models,” Proceedings of the Fifth International Conference on Rapid Prototyping, Dayton, Ohio, pp. 221~228, 1994.
6. B. Saelens, J. Pauwels, W. Vancraen, “Support generation for rapid prototyping,” Proceedings of the Sixth International Conference on Rapid Prototyping, Dayton, Ohio, pp. 115~121, 1995.
7. 허정훈, 이건우, “SLA를 이용한 신속시작작업을 위한 지지대 자동 생성 시스템의 개발,” 대한기계학회 추계학술대회논문집(I), pp. 788~794, 1995.
8. P.F. Jacobs, “Rapid Prototyping & Manufacturing: Fundamentals of Stereolithography,” Society of Manufacturing Engineers, First Edition, 1992.
9. G. Fadel, D. Crane, L. Dooley, R. Geist, “Support Structure Visualization in virtual Reality,” [http://www.vr.clemson.edu/dmg/papers/VR\\_papers.html](http://www.vr.clemson.edu/dmg/papers/VR_papers.html).
10. M. Bablani, A. Bagchi, “Quantification of Errors in Rapid Prototyping Processes, and Determination of Preferred Orientation of Parts,” Transactions of the North American Manufacturing Research Institution of SME, Vol. XXIII, pp. 319~324, 1995.
11. Guide to MAGIC S 2.2, Materialise N.V., 1993.