



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

선박 부재 형상의 3D모델링 및
부재 절단 시스템의 적용에 관한 연구

3D Modeling of Structural Members of Ship
and Cutting Macro System Development



지도교수 박 주 용

2009년 2월

한국해양대학교 대학원

해양시스템공학과

윤 정 열

本 論 文 을 윤 정 열 의 工學碩士學位論文으로 認准함.

위원장 : 공학박사 박 명 규 (인)

위 원 : 공학박사 박 주 용 (인)

위 원 : 공학박사 박 석 주 (인)



2009년 2월

한 국 해 양 대 학 교 대 학 원

목 차

List of Figures	II
Abstract	III
제 1장 서론	1
1.1 연구배경	1
1.2 연구 내용 및 목표	2
제 2장 도형의 생성	3
2.1 컴퓨터 그래픽스	3
2.2 물체의 표현 방법	4
2.3 모델의 생성 및 조작 방법	5
제 3장 Scene Graph를 이용한 도형의 생성	8
3.1 컴퓨터 그래픽 라이브러리	8
3.2 Scene Graph	10
3.3 형상 부재의 Scene Graph	12
제 4장 3D 형상 모델링의 절단 매크로 적용	21
4.1 절단 매크로(Cutting Macro)	21
4.2 기존의 절단 매크로 시스템	22
4.3 3D 형상 모델링을 이용한 절단 매크로 시스템	23
제 5장 결론	27
참고문헌	28

List of Figures

Figure 2.1 Wire frame, surface, solid model example	4
Figure 2.2 Vertex, edge, face	6
Figure 2.3 Sweep	6
Figure 2.4 Boolean operation	7
Figure 3.1 Modular structure of Open CASCADE	8
Figure 3.2 Open Inventor architecture	9
Figure 3.3 Scene graph structure	10
Figure 3.4 Base web part scene graph	13
Figure 3.5 Web part scene graph	13
Figure 3.6 Basic flange part scene graph	15
Figure 3.7 Flange part scene graph	15
Figure 3.8 Scene graph of L-angle	16
Figure 3.9 3D modeling of L-angle	17
Figure 3.10 Scene graph of H-beam example 1	18
Figure 3.11 3D modeling of H-beam example 1	18
Figure 3.12 Scene graph of H-beam example 2	19
Figure 3.13 3D modeling of H-beam example 2	20
Figure 3.14 3D modeling of H-beam example 3	20
Figure 4.1 Cutting macro example	21
Figure 4.2 Cutting macro & 3D modeling	22
Figure 4.3 Sequence diagram of 3D modeling macro system	24
Figure 4.4 3D modeling macro system	25
Figure 4.5 Intelligent input system	25

3D Modeling of Structural Members of Ship and Cutting Macro System Development

Jeong-Yeol, Yoon

**Division of Ocean Systems Engineering
Graduate School of Korea Maritime University**

Abstract

3D geometric modeling has a lot of advantages in the fields of design and manufacturing. Many manufacturing process and production lines are using 3D geometric modeling and virtual products. It helps to reduce cost and time.

Macro system is widely used in cutting and welding process to improve productivity. Conventional macro input system is using 2D text. Although its simplicity and promptitude have some benefits, it can sometimes lead to wrong product. In this paper, the conventional text macro input system is improved by addition of 3D geometric modeling technique. It is easy to find the faults to be able to occur before or during the cutting operations and fix it.

The 3D geometric modeling of structural parts of a ship was carried out by scene graph method which makes very complex shape simply construct. The scene graph method is implemented by Open-Inventor which is an object-oriented, cross-platform 3D graphics toolkit for the development of industrial-strength interactive 3D graphics applications.

제 1장 서론

1.1 연구배경

컴퓨터 기술의 발달과 기계 성능의 발전으로 인하여 산업 현장의 환경에 많은 변화가 생기게 되었다. 발달된 기계 시스템의 도입으로, 적은 노동력으로도 대량 생산을 할 수 있게 되었고, 생산 시간의 단축으로 인한 이윤 창출도 그만큼 증가 하게 되었다. 그러나 능동적이지 못한, 수동적인 기계의 힘만으로 작업을 할 수 있는 영역은 극히 드물다. 따라서 대부분의 기계들은 사람의 손을 거쳐야만 작업이 가능하다. 요즘 무인 비행기, 무인 자동차등 인공지능을 탑재한 제품들의 개발이 활발하게 이루어지고 있기는 하나, 아직까지 산업현장에서 이를 적용하기란 쉽지가 않다. 특히 조선 산업현장의 경우는 더더욱 그렇다. 다른 어느 곳보다도 발달된 기계 시스템을 도입하고는 있으나, 대부분 사람의 손을 거쳐야만 작업이 가능하다. 용접이나 절단의 경우 발달된 기계시스템을 도입하여 시간 단축과 생산량 증가를 목표로 여러 가지 연구가 이루어지고, 실질적으로 도입도 하고 있으나, 완전 자동화 기술을 이용하기란 어렵다. 하나의 강재가 들어온다 하더라도, 때에 따라 다른 형태의 모양을 절단 하거나 용접해야 하는데 이는 작업자가 그때 마다 판단하여 다른 명령을 내려 주어야만 한다.

형강재의 자동 절단 경우도 이에 해당된다. 형강재의 자동 절단 시스템은 작업자에 의해 절단 경로를 입력 받고, 절단 로봇은 정해진 경로에 따라 절단을 시작하게 된다. 절단은 절단 룸 안에서 이뤄지게 되며, 한번 입력된 경로대로 움직이기 때문에, 절단이 완료되기 전까지는 절단 형상의 정확도 유무를 판별하기가 쉽지 않다. 따라서 절단 경로를 잘못 입력한 경우, 시간과 비용적인 측면에서 큰 손실을 가져 올 수밖에 없다. 절단 룸 안에 카메라를 설치하여 작업자가 절단 과정을 지켜 볼 수도 있으나, 이는 절단 과정 중에 잘못된 점을 발견하면 절단 과정을 바로 취소시킴으로써 시간의 낭비를 조금이나마 줄이는 방법뿐이 안 된다. 이를 해결하기 위해서는 보다 근본적인 방법이 필요 하다.

1.2 연구 내용 및 목표

로봇을 이용한 자동 절단 시스템의 경우 고열과 슬러그로 인하여 대부분 절단 룸 안에서 이루어지게 된다. 한번 로봇이 절단 룸 안에서 작업을 시작하게 되면, 작업 경로를 바꾸는 것이 쉽지가 않다. 완성된 형상을 확인하기 위해서는 절단 작업이 끝난 후 컨베이어를 통하여 절단 룸 밖으로 형강재가 빠져 나온 후에 가능하다. 이는 절단 로봇에 잘못된 정보가 입력되었는지의 여부가 절단이 끝난 후에 판별이 가능하다는 의미이다. 그렇다면, 절단이 끝나기 전에 절단 후의 형상을 미리 확인해 볼 수는 없을까? 물론 정보 입력이 오류 없이 제대로 입력되었다면 문제가 되지 않을 것이다. 그러나 형상 정보와 네스팅 정보를 사람이 직접 입력 하는 만큼 실수는 나오기 마련이다. 실수로 인한 시간과 비용의 낭비가 계속 누적 된다면, 결국은 큰 손실을 초래할 것이다.

그리하여 본 논문에서는 이러한 실수를 방지하기 위해 절단 작업 전에 컴퓨터 3D 형상 모델링을 통하여 절단 후의 형상을 사전에 확인해 볼 수 있게 함으로써, 작업 능력의 최적화를 실현해 보고자 한다. 또한 형강재의 3D 형상 표현을 위한 최적의 방법도 함께 찾아보고, 이를 어떻게 적용시킬 것인가에 대해 논해 볼 것이다.

제 2장 도형의 생성

2.1 컴퓨터 그래픽스

2.1.1 컴퓨터 그래픽스

컴퓨터 그래픽스란 컴퓨터와 하드웨어 및 특정 소프트웨어를 이용하여 도형 및 화상, 그림을 만드는 기술로 붓이나 연필, 포스터컬러 등 미술 및 디자인 재료로 이용하던 것을 컴퓨터와 입출력 장치를 이용하여 만드는 모든 행위의 총칭이다. 즉, 컴퓨터의 하드웨어와 소프트웨어를 활용하여 물체의 형상, 성분, 색채 등 각종 성분을 컴퓨터에 입력, 이미지 변환작업을 통하여 제3의 영상을 만들어 내는 컴퓨터의 기술로서 종래의 도안이나 설계건축에 제한적으로 사용되던 것을 영화, 광고 산업, 컴퓨터아트, 시뮬레이션, 산업디자인 등 예술과 산업 분야 전반에 꿈의 이미지를 만들어 내는 21세기의 표현매체로 그 효율성과 기술이 날로 발전하고 있다. 이는 사용되는 프로그램에 따라 다른 효과를 볼 수 있는데, 크게는 정지화상과 동화상, 2D, 3D, 애니메이션으로 구분을 할 수 있다.

2.1.2 컴퓨터 그래픽스의 특징

컴퓨터 그래픽스의 대표적인 특징 4가지는 다음과 같다.

- 1) 고도의 정밀성이 요구되는 그림을 작성하거나 이론으로 구해진 각종 자연 현상이나 물리적, 화학적 특성 등의 가시화된 시뮬레이션은 컴퓨터 그래픽스(Computer Graphics, 이하 CG)에 의하지 않으면 실현 불가능하기 때문에 CG를 이용할 필요성이 있다. CG를 이용하면 매우 많은 복잡하고 규칙적인 모양이나, 수학적으로 정확한 도형을 만들기도 하고, 작성된 같은 종류의 모형을 동시에 대량으로 복사하는 것이 가능하며, 적당한 기억매체에 보존해두는 것도 가능하다. 그리고 기억된 도형은 필요에 따라 디스플레이 장치로 출력되며 또 프린터에 의한 인쇄도 할 수 있다.

- 2) 인간이 표출해 내고자 하는 아이디어의 세계를 시각화하여 실존하지 않는 공간과 화상을 전개 할 수 있으며 3차원 물체를 수작업이나 모델을 이용하였을 때보다 더욱더 자유롭게 재질과 색을 마음대로 변형시킬 수 있다.
- 3) 주변의 배경을 마음대로 조절할 수 있으며, 물감을 사용할 때 나타나는 한계적인 색상이나 질감, 휘도 등을 의도대로 표현할 수 있다. 실제로 나타낼 수 없는 부분까지 나타낼 수 있으며 2D의 경우 단시간 내에 아주 많은 양의 그림을 처리할 수 있다.
- 4) 대상물의 형태를 의도대로 변화시킬 수 있으며, 디자이너의 감각과 창조성을 확대하여 무한한 이미지의 창출을 가능케 하며, 시행착오를 줄이고 정확한 의사결정으로 제작시간의 단축을 이용, 기업의 원가 절감에 일익을 담당한다.

2.2 물체의 표현 방법

물체의 표현 방법으로는 대표적으로, Wire frame model, surface model, solid model 등이 있다.

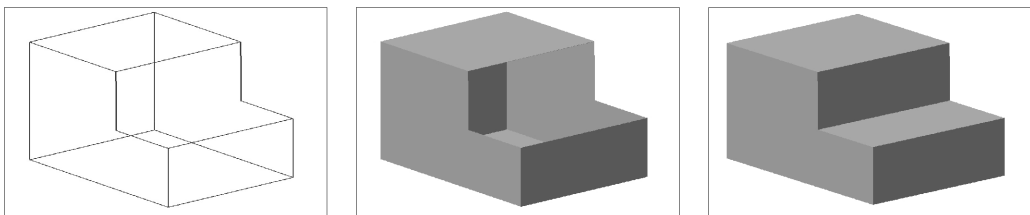


Figure 2.1 Wire frame, surface, solid model example

1) Wire frame model

Wire frame model은 3차원 물체를 모서리와 꼭지 점을 이용하여 표현하며, 모서리의 집합으로 구성된 것으로 간주할 수 있다. 이 모델은 다른 모

텔들에 비하여 비교적 메모리 공간이 적게 소요되고 쉽게 표현이 가능하지만, Surface에 대한 정보를 가지고 있지 않기 때문에 물체의 체적, 표면적, 무게중심 등의 정보를 얻을 수 없다. Wire frame으로 표현된 물체는 그 해석이 명확하지 않는 경우가 빈번히 발생하게 되는 단점이 있다.

2) Surface model

Surface model은 Wire frame model의 데이터에 Surface 데이터를 추가로 가지고 있는 모델이다. 단순히 표면의 집합으로 물체를 표현하기 때문에 3차원 물체에 대한 정보를 충분히 보유하지는 않는다.

3) Solid model

Solid model은 3차원 물체에 대한 정보를 가장 많이 표현하고 있다. Solid model에서는 Surface model에서 비어 있는 공간에 대한 정보를 보유하고 있기 때문에 설계분석 및 생산 공정의 많은 과정에 필요한 데이터를 자체적으로 산출할 수 있다. Wire frame model이나, Surface model에 비하여 복잡한 자료 구조를 보유하고 있으며, 모델의 수정이나 화면 디스플레이에 비교적 많은 시간이 소요되어 이 들 문제 해결에 해단 여러 연구가 진행되고 있다.

2.3 모델의 생성 및 조작 방법

본 논문에서 3D graphic으로 표현하고자 하는 L-형강, H빔 등은 다양체들로 이루어진 복합 다양체 모델들이라고 말할 수 있다. 복합 다양체 모델을 생성하고 조작하는 방법은 다음과 같이 3가지의 방법들이 있다.

첫 번째 방법은 vertex, edge, face 등의 위상요소의 수준에서 모델의 형상을 생성하고 수정하는 것이다. 이 방법은 모델을 조작하기 위한 가장 기본적인 방법이나, 실제 사용자가 직접 조작하기에는 어려움이 따른다.

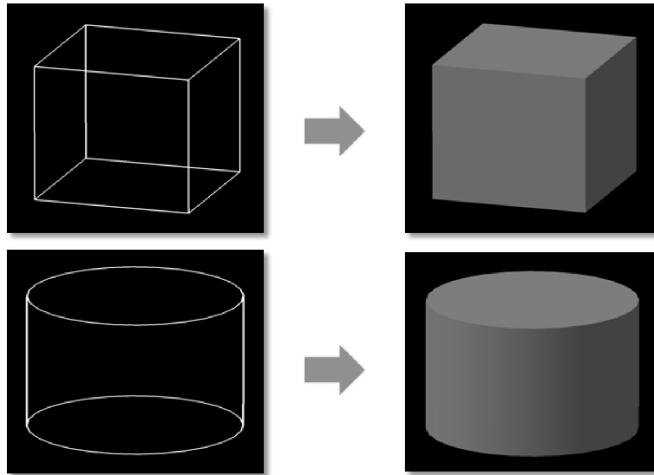


Figure 2.2 Vertex, Edge, Face

두 번째 방법은 점, 곡선, 곡면 등을 경로에 따라 움직이는 Sweep작업을 이용하여, 모델의 형상을 생성하는 것이다. 이 방법은 경로를 따라 일정한 단면 형상을 생성하는데 효과적이다. 특히, 복합다양체 자료구조에서는 같은 차원의 요소를 Sweep 하는 것도 가능하기 때문에 더욱 효과적이라고 할 수 있다.

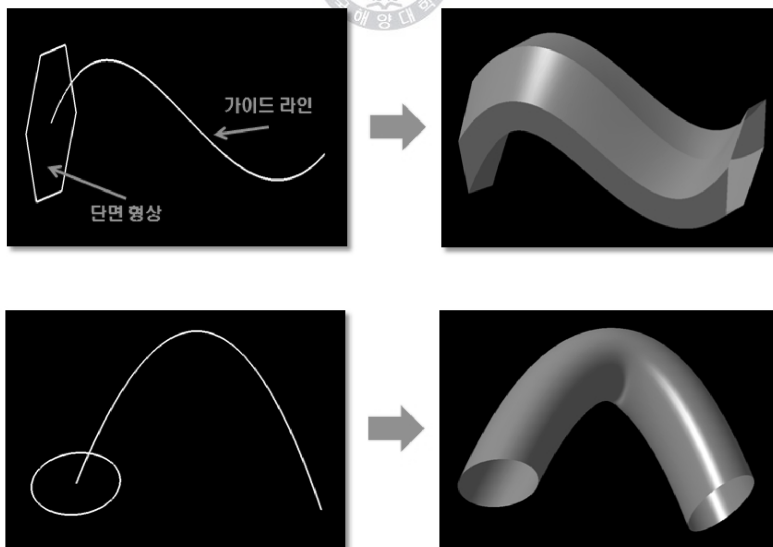


Figure 2.3 Sweep

세 번째 방법으로는 Merge, intersection, subtraction 등의 Boolean operation 을 이용하는 방법이다. 서로 다른 차원의 모델 사이에서의 Boolean operation 이 가능하고, 단순한 모델로, 복잡한 모델을 표현하기가 용이하다.

위 세 가지 방법 중에서, 복잡한 모델을 형성하기 위하여 단순한 모델을 먼저 생성한 후 이들 간의 Boolean operation을 통하여 원하는 모델을 얻는 것이 용이하고 작업에 소요되는 시간도 단축할 수 있는, Boolean operation을 이용한 형상 모델링 방법이 가장 많이 이용되고 있다.

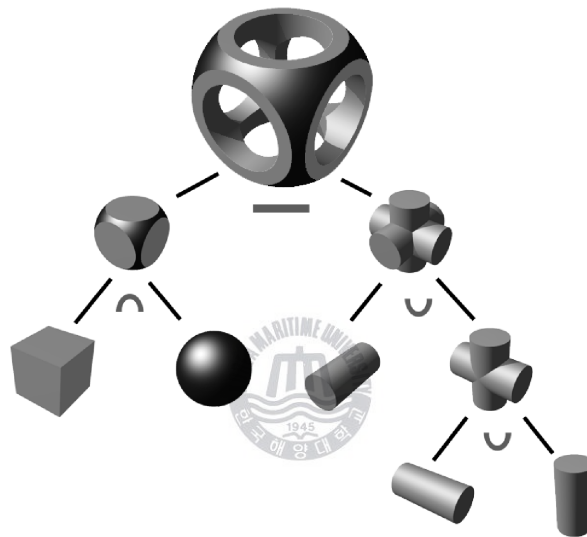


Figure 2.4 Boolean operation

제 3장 Scene Graph를 이용한 도형의 생성

3.1 컴퓨터 그래픽 라이브러리

3.1.1 Open CASCADE

Open CASCADE 는 3D CAD, CAM, CAE용 소프트웨어 개발 플랫폼으로서 개방형 CAD, CAM, CAE 커널이다. 커널이란 각각의 시스템 마다 계산 방식이 다른 수학 공식의 집합이고, 자동차와 비교 한다면 엔진과 같은 핵심 부분이라고 말할 수 있다. Open CASCADE는 과학적이고 기술적인 어플리케이션 개발을 위한 Open source 소프트웨어 컴포넌트이며, 모델링 기술을 위한 일련의 C++ 라이브러리이고 멀티 플랫폼(Linux, Windows, Solaris)을 지원한다. Open CASCADE는 여러 타입의 오브젝트들을 2D와 3D 기하학 정보로 모델링 할 수 있는 Toolkit이며, 오브젝트들을 디스플레이 하기 위한 가시화 서비스, 비 기하학적 데이터와 기하학적인 데이터의 결합, 표준 파일포맷(IGES, STEP)들의 Import와 Export를 통한 데이터 교환 서비스를 제공하므로 개발자가 보다 빠르게 어플리케이션을 개발하도록 도와준다.

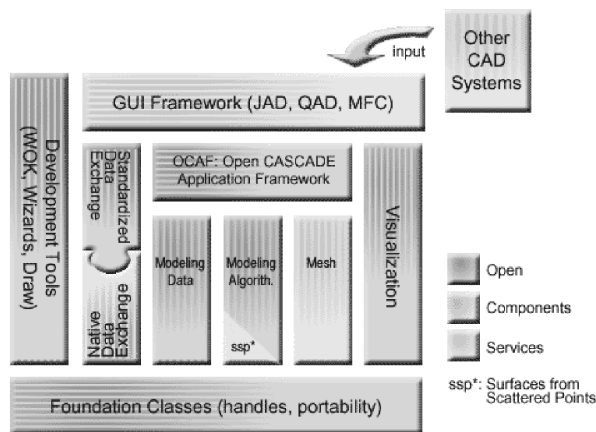


Figure 3.1 Modular structure of Open CASCADE

3.1.2 Open Inventor

Open Inventor는 그래픽 개발 유저들이 많이 사용하고 있는 객체 지향의 3D API toolkit 이다. Open Inventor의 실시간 상호 작용적인 프로그래밍 기법을 통해 개발자는 쉽고 빠르게 프로그램을 개발 할 수 있고, 운영체제나 기타 여러 시스템 환경에 영향을 받지 않은 높은 호환성을 가지고 있으며, VRML의 모태인 Scene graph database를 통해 직감적으로 오브젝트를 처리, 가공 및 변형할 수 있다. Scene graph상에서 생성되는 Object 들은 Node라는 개념에 의하여 구성되어 있고. Node의 정의 방법으로는 Method, Field, File format으로 구분되어 진다. Node의 종류로는 Group, Non-group으로 나누어지고, Group의 종류로는 Group, SoSeparator와 같은 하위 객체들을 가질 수 있는 Node들이 있고, No-group의 종류로는 하위 객체들을 가지지 않은 Shape, properties, cameras, lights와 같은 속성을 가지는 node들이 있다.

3D형상을 구현하기 위한 그래픽 라이브러리로 본 논문에서는 앞에서 살펴본 Open CASCADE 와 Open Inventor 중에 Open Inventor를 이용하여 3D 형상을 구현하였다.

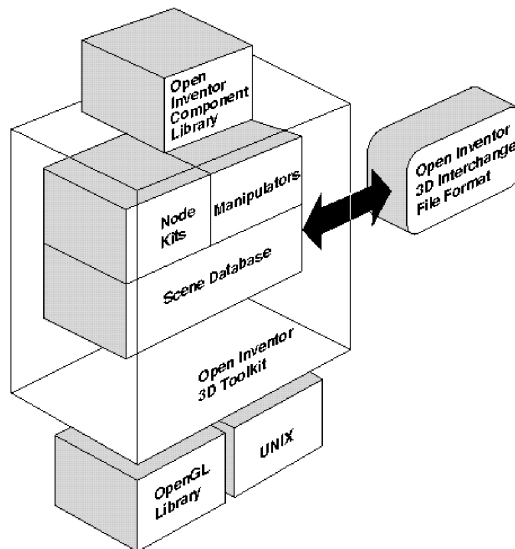


Figure 3.2 Open Inventor architecture

3.2 Scene Graph

3.2.1 Scene Graph

Scene graph는 Parent node들이 Child node들에게 영향을 미치는 곳에서의 상속에 대한 데이터를 정렬하는 방법이다. OpenGL과 같은 저수준 그래픽 라이브러리들은 Immediate mode 기반인 반면, Scene graph는 Retained mode 기반으로 데이터는 한번만 전달되고 필요할 때만 갱신된다. Scene graph가 관리하는 데이터는 그래프로서 구조화된다. Node는 링크에 의해 서로 연결되며, Root node는 Scene data의 시작점을 정의하는데 사용되어 진다.

Scene graph는 Open Inventor를 비롯하여 Java 3D, VRML 등 다양한 고차원 API에서 적용되어진 구조이며, 적은 용량에 비해 높은 성능을 보여주어 3D 형상 모델링에 많이 사용되고 있으며, 잘 알려진 Scene graph로는 OpenSG, Open Inventor, PLIB, SGL, OpenRM, Open Scene Graph, Perfomer 등이 있다.

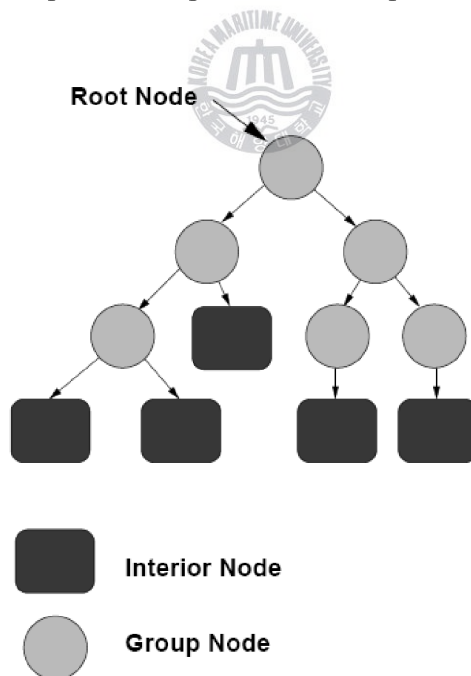


Figure 3.3 Scene graph structure

Figure 3.3은 Scene graph의 기본 구조를 나타내고 있다. Scene graph와 수학적 그래프의 차이는 Scene graph들은 여러 가지 다양한 요소들을 포함하고 있다는 것이다. 즉 노드들은 서로 다른 유형이다. Interior와 Group node 사이에 눈에 띄는 차이가 있다. Group node는 가시적인 기하도형을 담고 있으며, Interior node는 그래프를 논리적 그룹으로 구조화한다.

3.2.2 Scene Graph의 장점

Scene graph를 사용하게 되면 성능, 생산성, 이식성이 향상된다. Scene graph는 그래픽 처리를 최대로 하기 위한 기본 틀이기 때문에 이를 사용함으로써 여러 장점을 얻을 수 있는 것이다.

1) 성능

Scene graph는 그래픽 실행을 최대로 하기 위한 기본 틀과 Texture 와 Material을 같은 속성으로 정렬시키고 화면에 보이지 않는 객체를 Culling 하는 기술을 제공한다. CPU의 사용을 줄이고 GPU로 Culling 하게 되면 형상 표현을 위해 요구되어지는 방대한 자료의 양에 의해 수많은 작업을 하게 된다. Scene graph의 계층적인 구조는 이러한 과정을 매우 효율적으로 만든다. CPU의 사용 없이 GPU 만으로 Culling 하는 것은 속도를 증가시킬 수 있기 때문에 원활한 GPU의 작업을 위해 Scene graph의 중요성이 점차 부각되고 있다.

2) 생산성

Scene graph는 고품질의 형상 모델링을 할 때 요구되어지는 많은 작업을 줄일 수 있다. Scene graph는 수천 라인의 코드로 해야 할 일을 몇 줄의 코드로 관리할 수 있으며, OOP(Object Oriented Programing)의 개념으로 Scene graph tree 구조를 설계하면 재사용성이 증대되어, 형상 모델링에

필요한 유틸리티 개발 등에도 적용되어질 수 있다. 코드의 감소 화는 개발 비용 및 기간을 줄일 수 있는 직접적인 요소를 제공한다.

3) 이식성

Scene graph는 그래픽을 만드는 저수준의 작업을 캡슐에 싸고, 그것을 코드화 하여 플랫폼 특성을 감소시키거나 제거할 수 있다. Scene graph가 적용된 엔진을 이용할 수 있는 플랫폼에서 개발할 때 Scene graph의 코드는 플랫폼 특성을 타지 않으므로 타 플랫폼으로의 전환 시 많은 시간을 줄일 수 있으며, 플랫폼 전용의 코드를 사용하지 않을 경우 소스 코드를 재 컴파일 하는 단순한 작업으로도 전환이 가능할 수 있다.

3.3 선박 부재의 Scene Graph

본 절에서는 지금까지 살펴본 Scene graph를 이용하여 선박부재에 대한 3D 형상을 구현하였다. 3D 형상으로 구현되는 선박 부재는 L-형강과 H-beam을 그 대상으로 하였다.

3.3.1 Web 형상의 Scene Graph

L-형강은 1개의 Web과 1개의 Flange로 구성되어진다. 기본 Web 형상은 6개의 Surface 로 이루어진 하나의 Solid 형식의 직육면체로 표현할 수 있다. 일반적으로 Web과 Flange는 원형 그대로 쓰일 때 보다, 때에 따라 기본형상에서 일부분이 잘려져 나간 형태로 많이 쓰이게 된다.

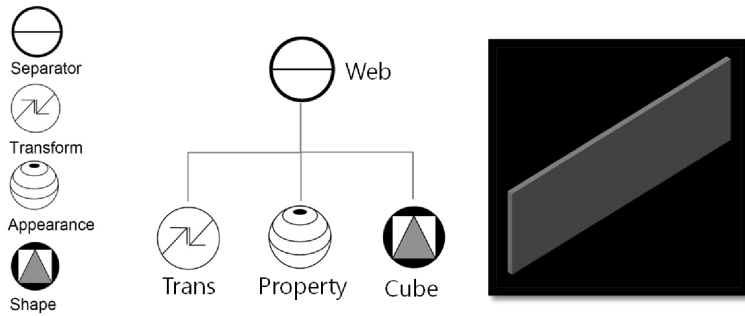


Figure 3.4 Base web part scene graph

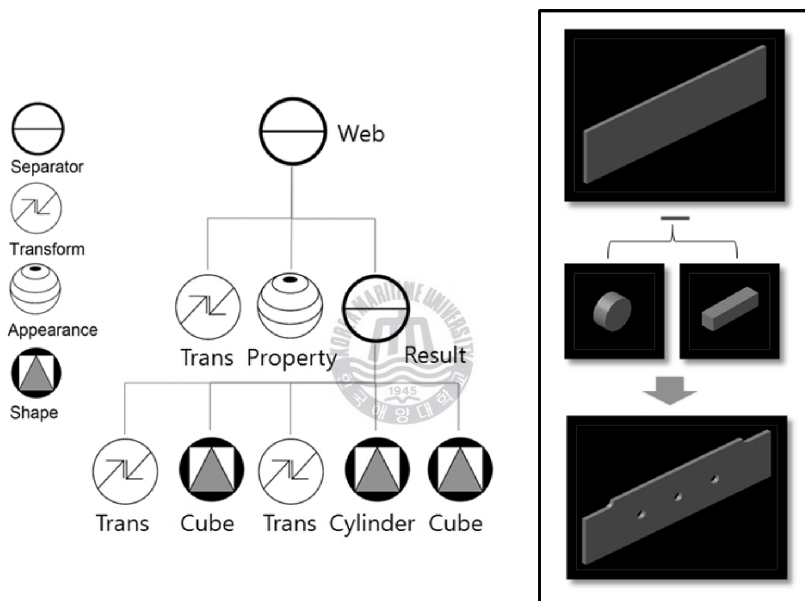


Figure 3.5 Web part scene graph

Figure 3.4 와 Figure 3.5 는 실제 사용되어지는 L-형강의 Web 형상을 나타낸 것이다. Figure 3.4는 기본적인 Web part를 나타낸 것으로 L-형강의 기본 Web 에서 아무런 변화가 없는 형태이다. Figure 3.5는 양쪽 끝단의 상단 부분이 잘려져 나가고, 가운데 3개의 홀이 뚫려 있는 형태의 형상을 나타내고 있다.

Figure 3.5에서 Parent node인 Web은 Child node인 Trans, Property, Separator 등으로 구성되어 있고, Separator node는 Trans, Shape node들을 Child로 가지고 있다. Cube 와 Cylinder 는 각각의 속성에 맞게 구조화 되어

있고, Solid 타입으로 이루어져 있다. 원하는 Web 형상을 얻기 위해서는 기본 형상인 2개의 Cube와 1개의 Cylinder가 필요 하다. 이것들을 2장 2절에서 설명한 'Boolean operation을 이용한 도형의 생성방법' 을 이용하여 Figure 3.5와 같이 원하는 형태의 3D 형상을 얻어 낼 수 있다. 본 논문에서 사용한 Open Inventor 에서는 SoCSGGroup 을 통하여 Boolean operation 기능을 기본 제공해 주고 있으며, 일반적인 표현 방법은 다음과 같다.

- myCSGGroup->expression.setValue("A-B")
- myCSGGroup->setOperationType(SoCSGOperation::SUB)

기본 파트인 첫 번째 Cube에 Subtraction할 두 번째 Cube를 알맞은 위치에 Transform 시키고 SoCSGOperation::SUB 을 이용하여 잘려진 형태의 형상을 만든다. Cylinder도 위와 마찬가지로 알맞은 위치에 Cylinder를 Transform 시키고 Subtraction 하여 최종적으로 원하는 형태의 형상을 만들 수 있다. 최종 형태의 Web 형상을 보면, 양쪽 끝단 부분과 가운데 홀 부분 등 Cylinder가 5번 사용되었으나, Figure 3.5에서 보면 Cylinder를 1개만 불러온 것을 볼 수가 있다. 이는 Scene graph를 이용함으로써 Cylinder 형상을 구조화 하여, 한번만 등록 시킨 후 원하는 위치에 단지 Transform 시킴으로써 여러 번 정의 하고 형상을 구현해야 하는 코드의 낭비를 줄일 수 있게 된 것이다.

3.3.2 Flange 형상의 Scene Graph

Figure 3.6 과 Figure 3.7 은 L-형강의 Flange 형상을 나타내고 있다. Figure 3.6은 기본적인 Flange part를 나타낸 것으로 L-형강의 기본 Flange에서 아무런 변화가 없는 형태이고, Figure 3.7 은 양쪽 끝단의 하단 부분이 원통형으로 잘려져 있는 형태이다. Flange 형상의 기본 구현 방법은 위에서 설명 Web 형상의 방법과 동일하다. 원하는 형태의 형상을 구현하기 위하여 Flange 에서는 1개의 Cube와 1개의 Cylinder를 이용하였다. Flange 형상에서는 양쪽 끝단에만 Cylinder를 이용하여 절단된 형상을 구현하였다.

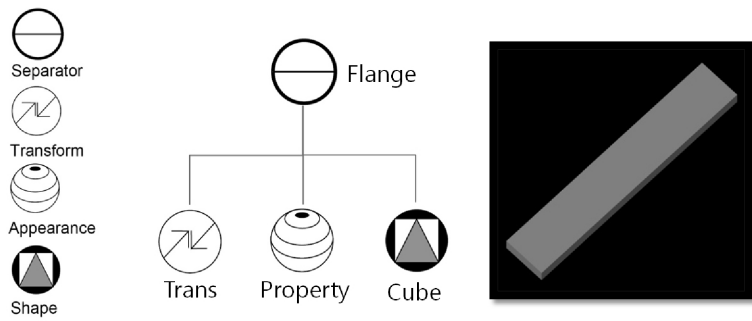


Figure 3.6 Basic flange part scene graph

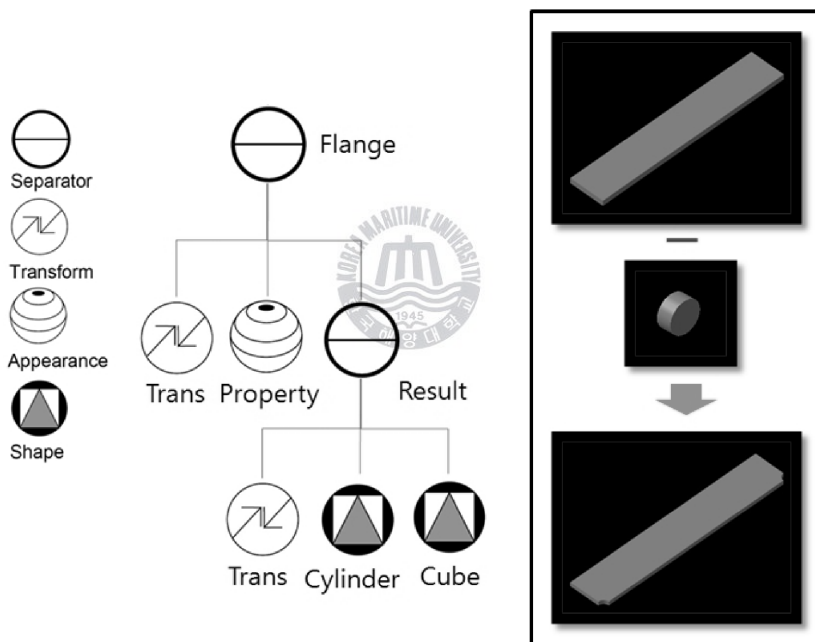


Figure 3.7 Flange part scene graph

3.3.3 전체 L-형강 형상의 Scene Graph

Figure 3.8 과 Figure 3.9 는 3.3.1과 3.3.2에서 설명한 Web과 Flange를 이용하여 최종적으로 만들고자하는 L-형강 형상을 나타내고 있다. 기본 형상인 Cube와 Cylinder를 원하는 위치에 Transform 시킨 후, Open Inventor에서 제공하는 Boolean operation인 SoCSGOperation을 이용하여 각각 원하는 형태의 Web과 Flange를 만들게 된다. Web과 Flange는 L-형강을 표현하기 위한 3차원 상의 알맞은 위치로 Transform되고, 최종적으로 이들을 Merge 시켜 원하는 형태의 L-형강 형상을 구현할 수 있다.

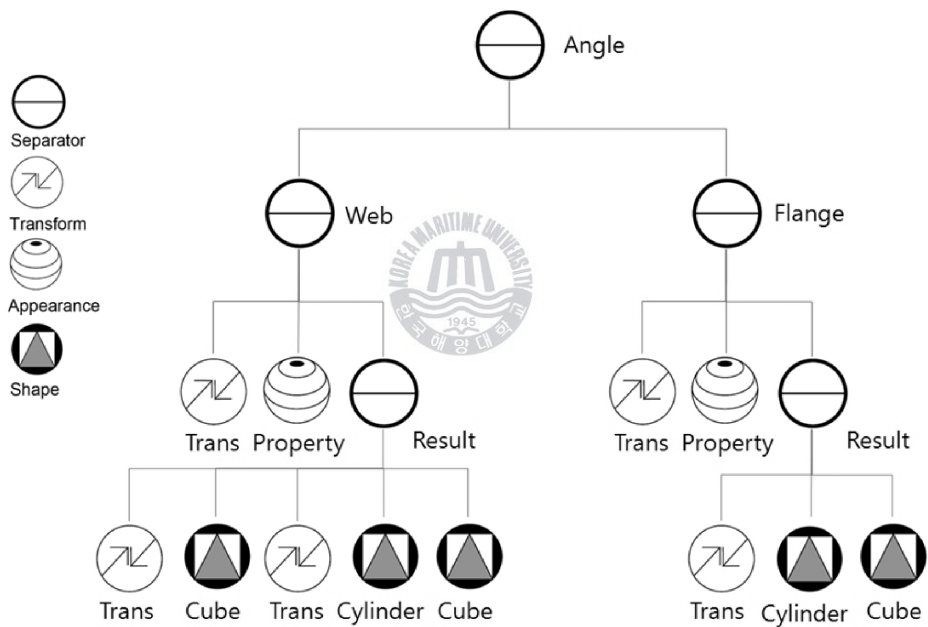


Figure 3.8 Scene graph of L-angle

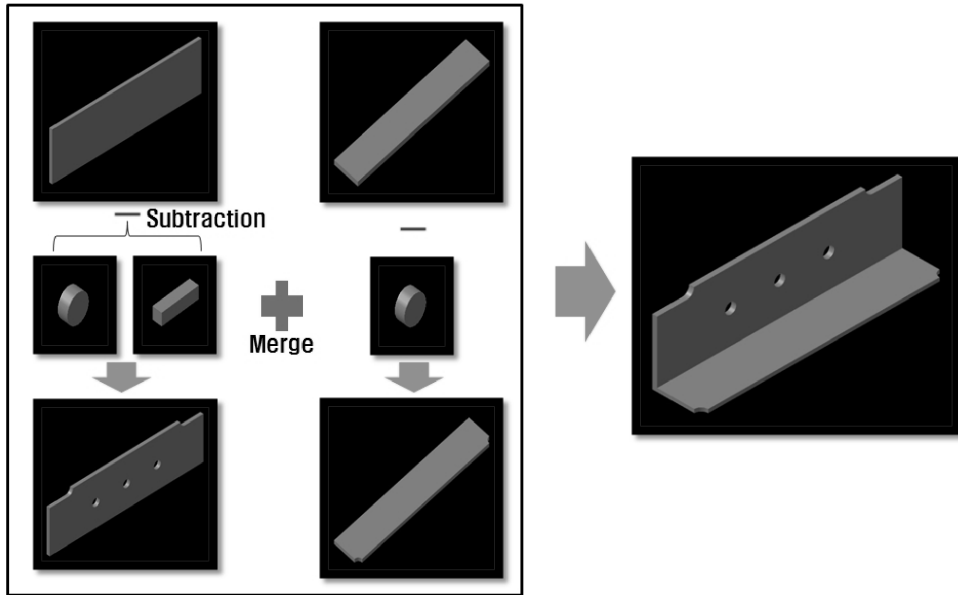


Figure 3.9 3D modeling of L-angle

3.3.4 H-beam 형상의 Scene Graph

H-beam은 Figure 3.10에 나타난 것처럼 2개의 Flange와 1개의 Web으로 구성되어 있다. 2개의 Flange는 각각 같은 형태의 형상일수도 있고, 혹은 서로 다른 형태의 형상일수도 있다. 도형을 생성하는 방법은 위에서 설명한 L-형강과 동일하다. H-beam에서는 Flange가 2개 이므로, 각각의 Flange의 형상이 동일한 경우와 서로 다른 경우에 형상을 구현하는 방식에 약간의 차이가 있다. Figure 3.10은 각각의 Flange 형상이 다른 경우를 나타내고 있다. Flange의 형상이 서로 다를 경우 각각의 Flange를 따로 구현해 주어야 한다. 따라서 Figure 3.10에서 Left flange와 Right flange를 따로 구현해 주었고, 그에 해당하는 형상은 Figure 3.11과 같다.

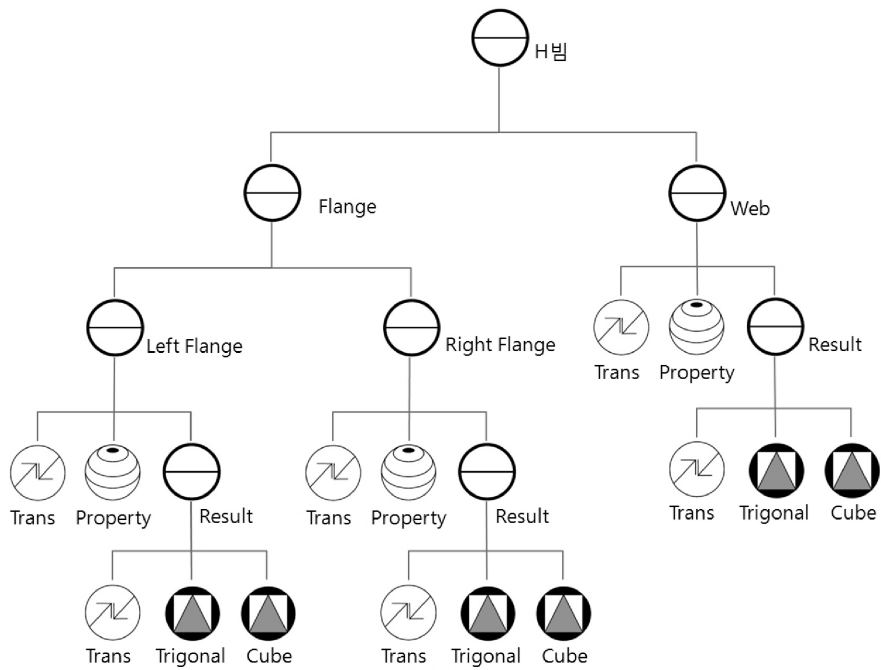


Figure 3.10 Scene graph of H-beam example 1

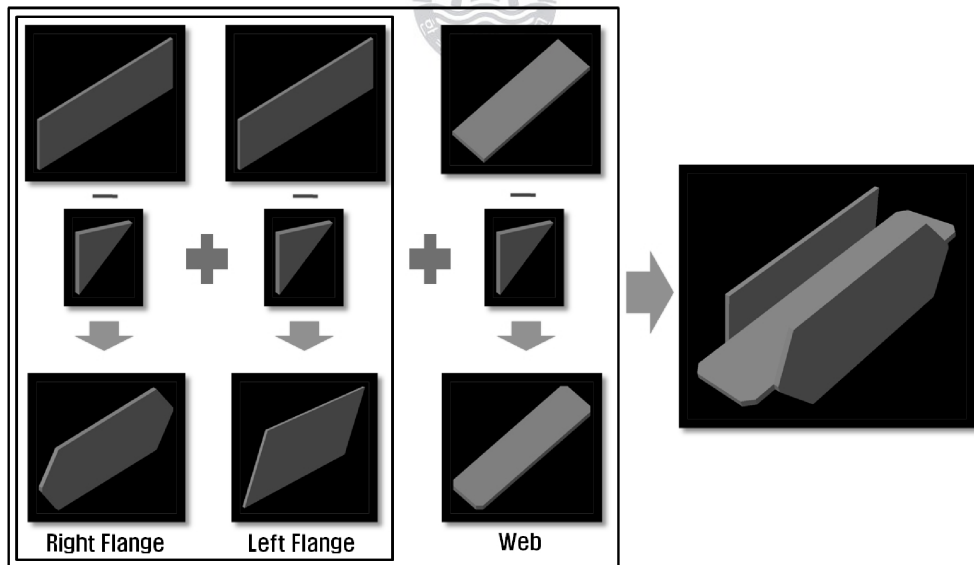


Figure 3.11 3D modeling of H-beam example 1

Figure 3.12는 Flange의 형상이 서로 같은 경우를 나타내고 있다. 두개의 Flange가 같은 형태의 형상을 나타낼 경우 Figure 3.10처럼 각각의 Flange를 따로 구현해 줄 필요가 없다. 먼저 하나의 Flange 형상을 구현해 주고 같은 형태의 다른 Flange는 단순히 Transform 시킴으로서 같은 형태의 형상을 표현해 줄 수 있기 때문이다. Figure 3.12에서 표현한 방법을 좀 더 자세히 살펴보면, L-형강과 마찬가지로 구조화 된 기본 형상인 Cube와 Trigonal을 알맞은 위치에 Transform 시키고 두 형상의 Boolean operation을 통하여 하나의 완성된 Flange 형상을 만들어 준다. 그 후 같은 형태를 가지는 나머지 Flange 형상을 단순히 Transform하여 3D 공간상에 표현하여 준다. 이렇게 하여 만들어진 2개의 플렌지와 같은 방법으로 완성되어진 Web 형상을 Merge 시켜 Figure 3.13과 같이 원하는 형태의 H-beam 형상을 구현할 수가 있다.

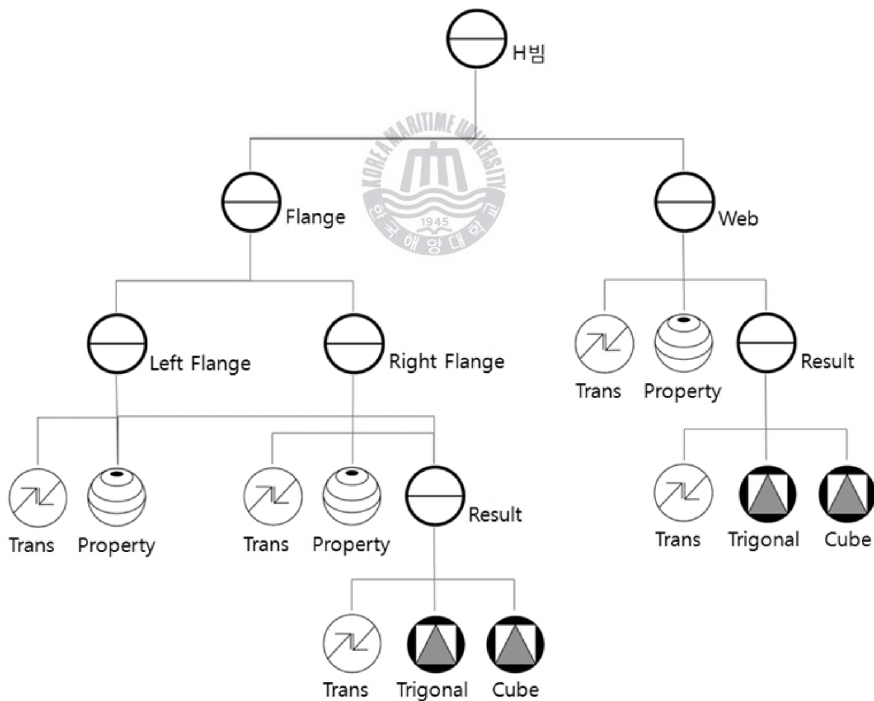


Figure 3.12 Scene graph of H-beam example 2

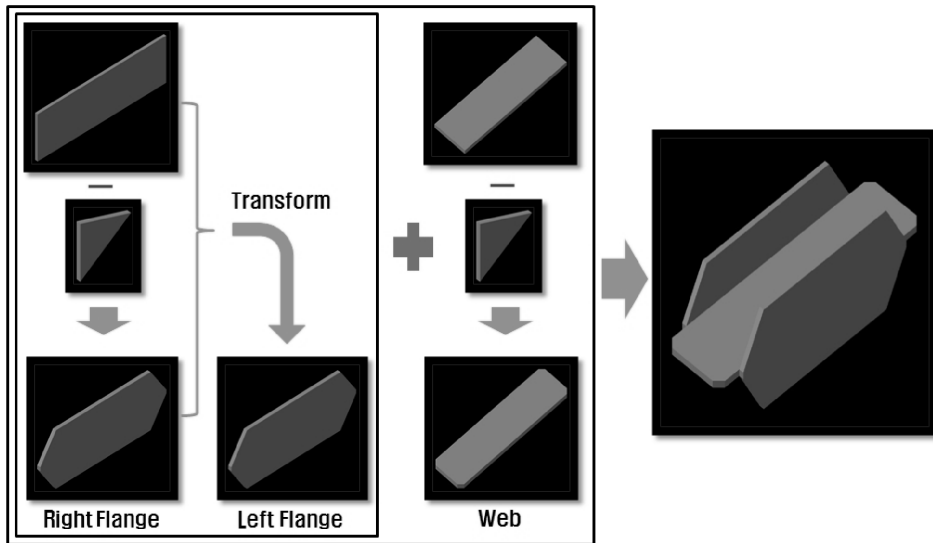


Figure 3.13 3D modeling of H-beam example 2

위에서 설명한 H-beam의 형상 구현 방법을 이용하면 Figure 3.14와 같이 Flange나 Web 부분에 Bevel이 들어간 형태의 형상이나, 좀 더 복잡한 형태의 형상을 쉽게 구현할 수 있다.

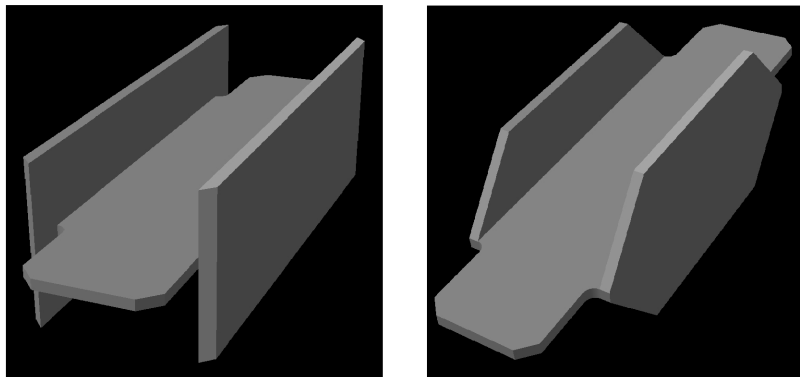


Figure 3.14 3D modeling of H-beam example 3

제 4장 3D 형상 모델링의 절단 매크로 적용

4.1 절단 매크로(Cutting Macro)

Macro input system에서는 부재 형상 절단경로를 절단 로봇에게 전달하기 위해 '매크로'라는 것을 사용한다. 매크로란, 어떤 특정 형상을 표현하는 약속된 문자들의 집합이며, 절단 매크로란, 절단 형상 모양을 입력하기 쉬운 약속된 문자로 표현한 것이다.

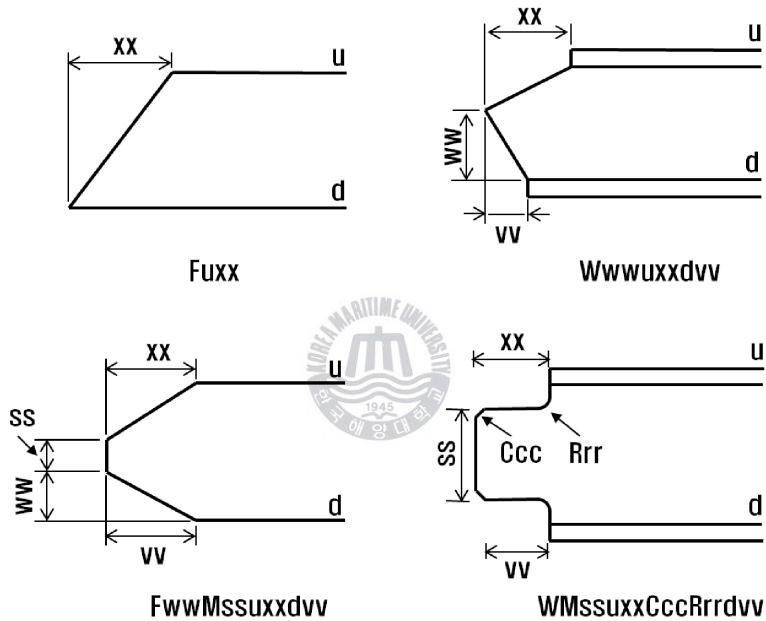


Figure 4.1 Cutting macro example

Figure 4.1 은 절단 매크로의 예이다. 먼저 Fuxx를 살펴보면, F는 Flange를 나타내며 u는 Flange의 상단, d는 Flange의 하단을 나타낸다. Fuxx는 Flange의 상단 쪽에 xx 만큼 잘려져 나간 형상을 의미한다. FwwMssuxxdvv 는 Flange에서 좀 더 복잡한 형상을 표현하고 있다. ss는 상단에서부터 잘려져 나가는 길이인 xx와 하단에서부터 잘려져 나가는 길이인 vv와의 교차되지 않은 중간 위치에서의 길이를 나타내며, 보통 ss의 중점은 Web 두께의 중점과 같은 위치를 가진다. 그리고 ww는 하단에서 부터 ss의 시작점까지의 길이를 나타내고 있다.

Wwwuxxdvv는 web의 상단방향으로 xx만큼, 하단방향으로 vv만큼 잘려 나가며, 잘려져 나가는 두 길이의 교차점이 하단부분에서부터 ww인 형태의 형상을 나타낸다. WMssuxxCccRrrdvv는 Web에서의 복잡한 형상을 나타내고 있다. ss는 Web의 길이에서 Flange의 두께와 상단과 하단에서 rr만큼 뺀 나머지의 길이를 의미한다. Ccc는 ss의 양쪽 끝점에서 길이와 높이 방향이 cc인 삼각형 형태로 잘려져 나간 형상을 나타낸다. 이처럼 매크로를 사용함으로써 복잡한 형상의 형태를 문자로 쉽게 표현할 수가 있다. Figure 4.2 는 절단 매크로에 따른 3D 형상을 나타내고 있다.

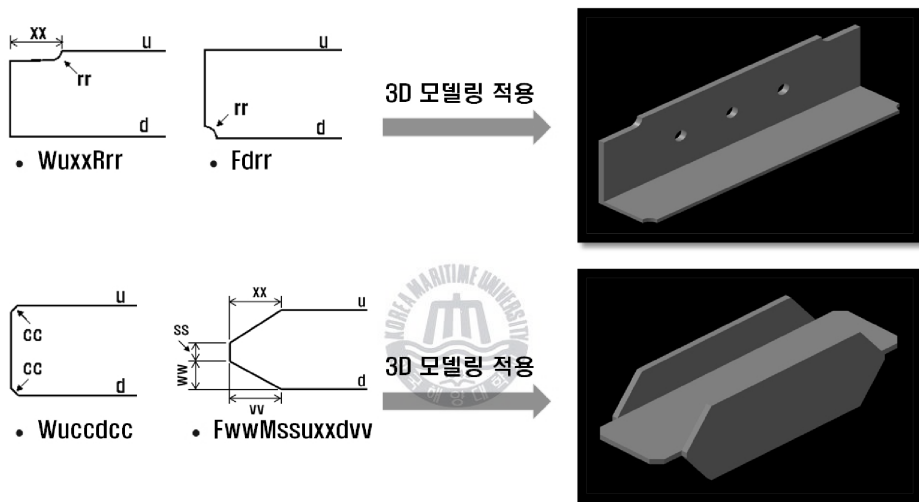


Figure 4.2 Cutting macro & 3D modeling

4.2 기존의 절단 매크로 시스템

4.2.1 텍스트 입력 위주의 절단 매크로 시스템

기존의 절단 매크로 시스템은 절단 매크로를 단순 텍스트 형식으로만 입력하는 시스템이며 현재 절단 매크로 시스템을 사용하는 대부분의 산업현장에서 사용되어지고 있는 방식이다. 캐드시스템에서 나온 형상 정보를 보고 작업자가 절단 매크로 시스템을 통하여 절단 매크로 정보를 입력하게 된다. 정보를 전달받

은 절단 로봇은 이를 토대로 부재를 절단 매크로에 맞게 절단하게 되는데, 로봇은 전달 받은 형상 정보가 알맞은 것인지 판단할 수가 없다. 작업자가 실수로 숫자를 잘못 입력 하거나, 네스팅을 잘못 켜올 경우에도 로봇의 작업영역 안에 존재만 한다면, 그대로 절단을 할 것이다. 작업자는 실수를 줄이기 위하여 절단 정보를 전송하기 전에 입력한 값들이 제대로 된 것인지, 재확인을 하기는 하지만, 영문자와 숫자로만 이루어진 복잡한 매크로를 가지고 형상의 모습을 추측해 내기란 여간 힘든 것이 아니다.

4.2.1 기존 절단 매크로 시스템에서의 문제점

한번 입력한 절단 형상 매크로를 다시 확인하면서 오류를 찾아내기란 쉽지 않은 작업이다. 반복적인 작업을 하다 보면, 입력하는 것에 대해 숙련도가 생기게 되고, 그로 인하여 재확인 및 검토 작업은 소홀해 지게 되기 마련이다.

절단 형상 매크로를 단순 텍스트 입력 방식으로 하였을 경우에 생길 수 있는 가장 큰 문제점은 입력된 절단 형상 매크로의 정확도 여부를 미리 확인 하지 못한다는 점이다. 절단 로봇이 입력 받은 정보들을 중심으로 절단 룸 안에서 작업을 하고, 컨베이어를 통하여 밖으로 내보내기 전까지는 실질적으로 형상 확인이 어렵고, 절단 룸 안에 카메라 등을 설치하여 실시간으로 절단 형상의 단면을 확인한다 하더라도 이미 때는 늦은 것이다. 한 부재에 대하여 각 파트들을 알맞게 네스팅하여 절단 로봇에 절단 매크로를 전송하게 되는데, 이중 하나를 잘못 절단 한다면, 해당 파트를 다른 부재에 다시 네스팅해 줘야 하며, 이로 인하여 부재의 손실과 시간의 낭비를 초래하게 된다.

4.3 3D 형상 모델링을 이용한 절단 매크로 시스템

앞에서 설명한 절단 매크로 시스템의 문제점을 보완하기 위해 본 논문에서 제시하는 방법은 다음과 같다. 캐드시스템에서 받은 정보를 데이터베이스에 담아두고 매크로를 입력할 때 마다 Intelligent input system을 통하여 해당 매크

로의 정확성 여부를 판단하고, 절단 형상 매크로를 절단 로봇에 전송하기 전에 해당 매크로의 절단 후 형상을 3D view system을 통하여 3D 그래픽으로 사전에 확인 하는 것이다. 3D 그래픽으로 미리 절단 후의 형상을 확인하게 되면, 작업자가 어떤 매크로를 입력하였는지, 그리고 어떤 부분에서 잘못된 부분을 입력하였는지 직관적으로 쉽게 파악할 수가 있다.

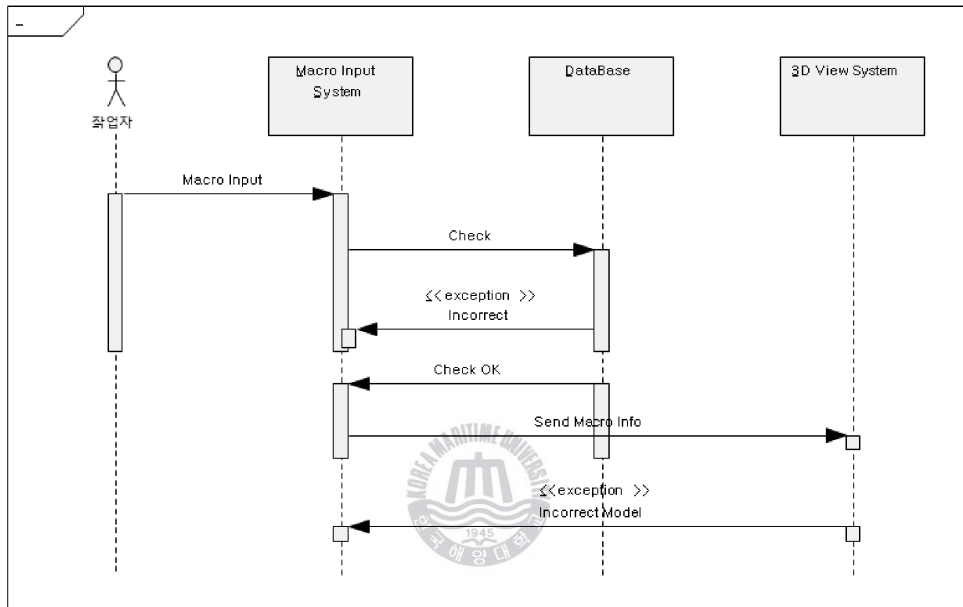


Figure 4.3 Sequence diagram of 3D modeling macro system

Figure 4.3은 3D Modeling macro system 의 Sequence diagram 을 나타내고 있다. 작업자가 절단 매크로를 입력하게 되면, Macro system과 Database가 서로 연동하여 매크로의 정확성 유무를 판단하게 된다. 입력된 매크로에 이상이 없을 시 3D view system을 통하여 절단 후의 형상을 작업자에게 보여 주게 된다. 작업자는 3D view system을 통하여 절단 후의 형상이 원하는 형상을 나타내고 있는지 확인 한 후에, 최종적으로 절단 로봇에게 절단 매크로를 전달하게 된다. 이로 인하여 작업자의 실수로 인한, 시간과 비용의 낭비를 줄일 수 있게 된다.

Figure 4.4 는 위에서 소개한 Intelligent input system과 3D view system을 적용하여 만든 3D modeling macro system 이다.

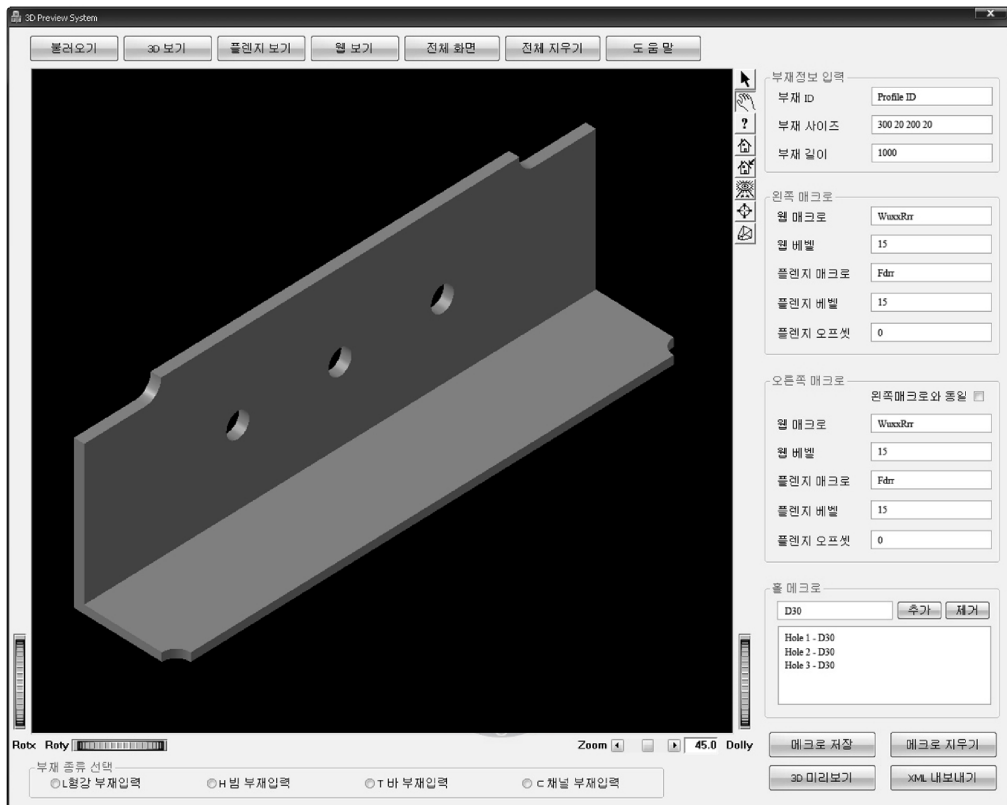


Figure 4.4 3D modeling macro system

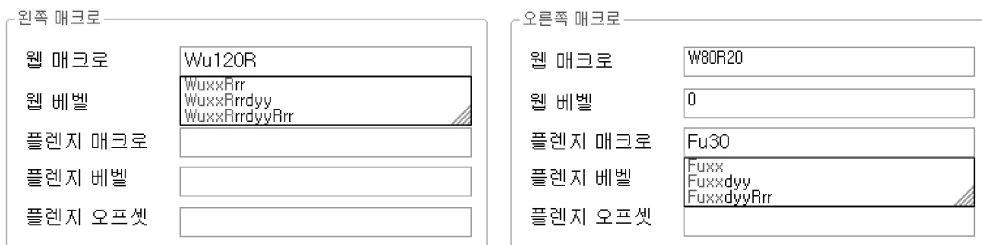


Figure 4.5 Intelligent input system

오른쪽 사이드 부분에 부재 정보와 절단 매크로를 입력하고 하단 부분에는 부재의 종류를 선택하게 된다. 절단 매크로 입력 시 Figure 4.5 의 Intelligent

input system이 적용되어 진다. 입력 가능한 절단 매크로를 데이터베이스에 등록해 두고, 작업자가 매크로 입력 시 입력 가능한 매크로를 팝업박스로 보여지게 된다. 잘못된 매크로를 입력하거나 입력이 불가능한 매크로를 입력 시 팝업박스가 사라지면서 사용자에게 에러 메시지를 띄우게 된다. 따라서 작업자는 매크로를 입력하는 동시에 매크로의 정확성 유무도 판별해 볼 수 있다. 입력하는 매크로에 이상이 없을 시 오른쪽 하단의 매크로 저장 버튼을 눌러 절단 로봇에게 매크로를 전달하기 전에 3D 미리보기 버튼을 이용하여 절단 후의 형상을 3D view 영역에서 미리 확인해 볼 수가 있다.



제 5장 결론

3D graphics를 적용하여 얻을 수 있는 장점은 무수히 많다. 공장을 건설하기 전에 설계도에 따라 미리 3D 형상으로 구현하고 시뮬레이션 하여 공장의 건설 과정이나 건설 후에 발생할 수 있는 문제점들을 파악하여 이를 사전에 방지 할 수 있다. 이 외에도 산업 전반의 제조 과정에서 3D 형상 구현을 통하여 제조 과정에서 발생 할 수 있는 문제점들을 미리 파악하여 이를 사전에 대처하고 발전 시켜 나가고 있다. 이렇듯 컴퓨터를 이용한 3D 형상 구현의 적용은 제조나 설비의 크고 작음에 관계없이 작업의 효율을 늘릴 수 있다. 본 논문에서도 이러한 3D 형상 구현을 통하여 얻을 수 있는 이점과 3D 형상을 구현하기 위한 효과적인 방법에 대하여 설명하였다.

기존의 텍스트 형식이로만 이루어지던 Macro 입력 방식에서 발생할 수 있는 문제점들을 파악하여 사전에 대처 할 수 있는 3D 형상 모델링을 이용한 절단 시스템 방법을 제시 하였다. 이 방법을 통하여 작업자는 절단 전에 절단 후의 형상을 미리 확인해 봄으로써, 입력 오류로 인한 부재의 손실 및 시간의 낭비를 최소화 할 수 있다.

형강재의 3D graphic 형상 구현을 위한 여러 가지 방법 중에서 가장 효과적인 방법으로 Boolean operation을 이용한 형상 구현 방법에 대해 설명 하였다. Boolean operation은 형상 모델링 방법 중 가장 많이 사용되는 방법으로써, 복잡한 모델을 생성하기 위하여 단순한 모델을 먼저 생성한 후 이들 간의 Boolean operation을 통하여 원하는 모델을 얻는 방법이다. 본 논문에서는 L-형강과 H-beam을 생성하기 위하여 Cube, Cylinder, Trigonal 등의 기본도형들을 구조화 하고, 이들 간의 Subtraction, Merge 등을 통하여 원하는 형태의 L-형강과 H-beam 형상을 나타내었다.

절단 과정뿐만 아니라, 위의 두 가지 방법들을 활용하여 용접 및 조립과정 등 비슷한 다른 분야에도 적용한다면 전체적인 작업 효율 증대를 꾀할 수 있을 것이다.

참고문헌

- [1] Josie Wernecke, <<The Inventor Mentor>> Addison-Wesley (1994)
- [2] Josie Wernecke, <<The Inventor Toolmaker>> Addison-Wesley (1994)
- [3] Richard S. Wright, <<OpenGL SuperBible>> 정보문화사 (2005)
- [4] 주우석, <<OpenGL로 배우는 컴퓨터 그래픽스>> 한빛미디어 (2006)
- [5] 이상현, 이견우, "비다양체 형상 모델링을 위한 간결한 경계 표현 및 확장된 오일러 작업자", 한국 CAD/CAM 학회 논문집, 제 1 권, 제 1 호, p1-19, 1996.3
- [6] 이동진(2005), "3D 블록의 조합에 의한 모델링과 3D Animation 에 관한 연구"
- [7] 유병현(1998), "선택저장 자료구조를 이용한 복합다양체 모델의 불리언 작업"
- [8] 전대원(2002), "Open Inventor를 이용한 이족보행로봇의 시뮬레이터 개발"
- [9] 이경상(2000), "3차원 표면 재구성을 위한 효율적인 다각형 감소 알고리즘에 관한 연구"
- [10] 정은수(2003), "장면 그래프를 이용한 3D 게임엔진의 설계에 관한 연구"