

컴퓨터 그래픽스 기법을 이용한 초고층 강제어초에 관한 연구 (I) - 설계지원 프로그램 개발

신현경*, 박호균**

A Study on Very High-rise Steel Artificial Reefs Based on Computer Graphics - part I (Design Aided Program Development)

by

Hyunyoung SHIN* and Hokyun PARK**

요 약

초고층강제어초(VERY High-rise Steel Artificial Reef, VEHSAR)의 형상 표현과 해석을 위해서 3차원 모델링 프로그램을 Windows 95 (또는 Windows NT) 환경하에서 MFC(Microsoft Foundation Class)를 기반으로한 Visual C++로 개발하였다. 컴퓨터를 통해 미리 초고층 강제어초 건조를 실제와 같이 구현해 봄으로써 문제점의 사전 예측 및 대책 수립 등을 가능케 하는 저비용 고효율의 일관설계시스템을 구축하는 것을 목표로 하는 이 프로그램은 객체지향프로그램 기법과 CAD/CAM기술의 도입으로 생산성 향상이 기대되며, OpenGL이라는 그래픽 라이브러리를 도입하여 좀더 빠른 고품질의 3차원 그래픽을 화면상에 나타나게 할 수 있다.

Abstract

The solid modelling program was developed for the 3-D representation and analysis of Very High-rise Steel Artificial Reefs(VEHSAR) using Visual C++ on the Microsoft's Windows 95 (or Windows NT) based on MFC(Microsoft Foundation Class). Importing oriented-object program and CAD/CAM technique, this integrated design system which aims at low cost and high efficiency makes it possible to predict problems and to establish counterplans through visualizing the construction of Very High-rise Steel Artificial Reefs in advance. Also it is useful in enhancing productivity. Introduction of OpenGL makes it display high quality 3D graphics more faster.

발 표 : 1998년도 대한조선학회 춘계학술대회('98. 4. 17)

접수일자 : 1998년 5월 29일, 재접수일자 : 1998년 10월 12일

* 정회원, 울산대학교 수송시스템공학부

** 학생회원, 울산대학교 수송시스템공학부

1. 서 언

21세기 선진수산업기술개발을 위해, 중요한 차세대 수산 증식 구조물의 하나인 초고층 강제어초(Very High-rise Steel Artificial Reef, VEHSAR)의 설계 및 건조 능력의 확보와 수주에서 인도까지의 모든 공정을 일목요연하게 파악하고 관리하기 위한 일관설계시스템의 필요성이 제기되고 있다.

국내에서는 “연안어장의 생산성 향상”을 위해 각종 인공어초가 개발되어 왔으나, 주로 콘크리트 재질의 소형어초가 제작·침설되었고, 강제어초는 그 장점에 비해 개발속도가 매우 느린 편이다. 1976년 이래 일본에서는 강제어초를 실용화하기 시작하여 1994년도에는 약 1200기의 강제어초가 제작설치되었다. 가공성이 뛰어나 복잡한 구조·형상의 어초제작에 적합하고, 고강도이기 때문에 대형고층구조물로의 제작에 의한 표·중층성어류의 유집효과를 기대할 수 있으며, 대수심에도 적용이 가능한, 초고층강제어초의 설계·제작은 국내 연안어장의 생산성 향상을 위해 시급한 과제이다.

모든 산업분야에서 신기술개발과 CIM을 통한 생산성 향상에 주력하여 저비용 고효율 생산체제를 정착시키려 노력하고 있고, 또한 많은 공학분야에서 컴퓨터그래픽스 기법을 이용한 기술 개발에 노력하고 있는 이때에 초고층강제어초를 위한 일관설계시스템 개발은, 시기 적절하다고 볼 수 있으며, 선진수산업기술개발에 일조를 할 것이다.

높이가 30~40m정도인 초고층 강제어초는 하부 중앙에는 콘크리트 뼈대가 있고, 중앙부에는 산란장이 있으며, 상단부에는 고기를 회유시키는 장치가 있어 다양한 기능을 갖춘 획기적인 어초이다.

본 논문에서는 Windows95 (Window NT) 운영체제 하에서 Visual C++를 사용한 그래픽사용자 인터페이스(GUI)를 구현하였으며, 초고층 강제어초를 구성하면서, 또한 자주 사용되는 모듈(Module) 과 각종 보(Beam), 평판(Plate)등을 원형요소로 정의하여 형상정보를 결정하고, 하향식(Top-Down) 방식을 사용하여 강제어초의 위상정보를 결정하였다. 형성된 형상정보와 위상정보를

이용하여 강제어초의 각 부재를 쉽게 조합 배치할 수 있게 함으로써, 초고층 강제어초 전용 모델러(VEHSAR Modeller)를 개발하였다. 한편, 현재 과학, 엔지니어링 모델링, 영상화 응용프로그램의 산업표준으로 사용되고 있는 OpenGL을 적용함으로써 보다 향상된 그래픽 수행능력을 갖게 하였다.

그리고 파라메트릭 설계(Parametric design)와 재 설계(Redesign)기능을 추가하여 반복되는 작업과정을 줄이고 설계변형이 가능하도록 하였다. 이를 통해 초고층 강제어초의 설계를 위한 기본자료를 쉽게 확보할 수 있으며, 3차원 형상정보 및 위상정보를 이용하여 자재물량정보 데이터 등을 생성할 수 있을 것이다. 또한 컴퓨터를 통해 미리 초고층 강제어초 건조를 실제와 같이 구현해 봄으로써 문제점의 사전 예측 및 대책 수립 등을 가능케 하는 저비용 고효율의 일관설계시스템을 구축하는 것을 목표로 하고 있다.

초고층강제어초 전용 VEHSAR Modelling 프로그램의 특징은 다음과 같다.

- ◎ Solid Modelling
- ◎ Top-Down Technique
- ◎ Graphical User Interface(GUI) 구현
- ◎ OpenGL 사용 [1~8].

2. 객체 지향 모델링

개발된 프로그램에 적용된 객체 지향의 개념은 다음과 같은 특징과 장점을 가지고 있다[2~3].

- 캡슐화(Encapsulation)
- 정보은닉(Information Hiding)
- 상속(Inheritance)
- 다형성(Polymorphism)
- 생산성 향상
- 유지보수용이

프로그램에서 사용되어진 class를 예로 설명하여 보자. class CPoint_3D는 아래와 같이 x,y,z 값을 데이터로 가지며, 데이터를 처리할 함수들로 구성되어 있다.

```
class CPoint_3D(
protected:
    double x;
    double y;
    double z;
public:
    CPoint_3D(void);
    CPoint_3D(double _x, double _y, double
_z);
    ~~~~~ 중략 ~~~~~
    CLineP operator=(CPoint_3D _point);
    ~~~~~ 중략 ~~~~~
    double Z(void) {return z;};
};
```

class CPoint_3D는 위상정보는 갖지 못하고 있다. 즉 기하하적인 정보만을 가지고 있으며 이 절점이 어떤 부재인지는 알 수 없다. 이의 해결을 위해 새로운 위상정보를 나타내는 class CGeoObject를 만들어서 이에 파생된 새로운 class를 만들고 CPoint_3D를 데이터로 가진다면 기존의 코드를 재사용할 수 있게 되는 것이다. CGeoObject는 name과 id라는 두 개의 데이터를 가지고 있다. name에는 부재의 이름이 포함되며, id에는 부재를 나타내는 위상정보가 들어간다.

```
class CGeoObject {
#define CGeoObject_SIZE 11
protected:
    char name[20];
    int id;
public:
    ~~~~~ 중략 ~~~~~
};
```

class CGeoObject를 상속받은 class CPoint_3D를 CGeoPoint_3D라 하자. 아래에서와 같이 두 개의 class를 적절히 조합하여 사용함으로 새로운 class를 생성한다. 그리고 객체지향프로그램의 특징을 사용함으로 보다 효율적인 작업이 가능하게 된다.

```
class CGeoPoint_3D : public CGeoObject {
```

```
public:
    CPoint_3D Point;
    ~~~~~ 중략 ~~~~~
};
```

3. OpenGL

SGI(Silicon Graphics Inc.)에서 배포된 도큐먼트에 따르면 OpenGL은 '그래픽 하드웨어를 위한 소프트웨어 인터페이스'라고 정의되어 있다. 이 말은 OpenGL이 3차원 그래픽을 위한 프로그래밍 언어이거나 3차원 모델을 만들고 렌더링하는 3D 스튜디오같은 그래픽 소프트웨어가 아님을 의미한다. OpenGL은 3차원 그래픽을 이용하는 응용프로그램을 만들기 위한 API이다. 이런 OpenGL은 점, 선,면 등과 같은 3차원 요소와 비트맵 등의 2차원 요소의 표현, 변환행렬을 통한 이들 요소의 변형, RGBA모델과 INDEXED COLOR모델에 의한 색상의 지원, 다양한 조명과 셰이딩의 설정, 텍스처 매핑 등 3차원 그래픽 라이브러리에 기대할 수 있는 거의 모든 것을 지원한다. 뿐만 아니라 앤티앨리어싱(Antialiasing), 안개효과, 블렌딩(Blending), 더블버퍼링(Double Buffering), 모션블러(Motion Blur), NURBS(Non-Uniform Rational B-Spline) 등의 고급기능도 구현할 수 있다. 본 프로그램에서 OpenGL의 특징인 더블버퍼링(Double Buffering)이라는 기능을 이용하여 그래픽 처리속도를 향상시켰다. Double Buffering이라는 것은 그래픽카드의 메모리를 두 개로 나누어서 데이터를 저장하는 방식으로서 OpenGL을 사용하지 않으면 컴퓨터의 메모리(Ram)를 사용하기 때문에 속도가 떨어지게 되는데 반면에 OpenGL을 사용하면 그래픽 카드의 메모리를 사용하기 때문에 속도가 향상 된다[9, 10].

본 프로그램에서 OpenGL을 사용하기 위해 만든 함수들을 예로 들어보자.

```
void GLResize(int cx, int cy)
{
    ~~~~~ 중략 ~~~~~
    glViewport(0, 0, cx, cy);
```

```

~~~~~ 중략 ~~~~~
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
~~~~~ 중략 ~~~~~
glMatrixMode(GL_MODELVIEW);
}
void GLSetupRC(void *pData)
{
    ~~~~~ 중략 ~~~~~
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_COLOR_MATERIAL);
    ~~~~~ 중략 ~~~~~
    glEnable(GL_LIGHTING);
    ~~~~~ 중략 ~~~~~
    glEnable(GL_LIGHT0);
    ~~~~~ 중략 ~~~~~
    glShadeModel(GL_SMOOTH);
}

```

GLResize()라는 함수는 윈도우의 클라이언트 영역이 너비와 높이를 전달하면서 WM_SIZE 핸들러에 의해 호출되는 함수이다. 여기서 뷰잉 볼륨과 뷰포트를 설정하기 위해 재계산이 필요할 수 있다. OpenGL에서 제공하는 glMatrixMode()라는 함수는 어떤 행렬 스택(GL_MODELVIEW, GL_PROJECTION)이 행렬 작업에 사용되는지 결정하는데 사용된다. 이 함수에 사용하는 행렬모드의 특징은 다음과 같이 GL_MODELVIEW는 장면에 오브젝트를 옮기는데 사용되고 GL_PROJECTION은 클리핑 볼륨을 정의하는데 사용된다.

그리고 GLSetupRC()라는 함수는 렌더링 컨텍스트를 초기화하는데 사용하는 함수이다. 이 함수에서는 바탕색상을 설정한다든지 명암상태를 설정하는 복잡한 작업들이 이루어지는 함수이다[9].

4. 하향식(Top-down) 개념

기존의 구조해석용 패키지에서 제공되는 모델링 프로그램들은 구조물을 정의하기 위하여 유한 요소적 접근으로 모델링하며 작은 것들의 집합으로부터 큰 것으로 변해가는 상향식(bottom-up)방식

을 취하고 있다. 즉 먼저 절점(node)을 생성하고 그 요소들에 재료의 특성을 부여한다. 이러한 작업을 반복함으로써 구조물들을 모델링한다. 반면 구조물 설계자들은 먼저 치수와 주요부 구조물(substructure)의 위치를 결정한 다음 부재들이 놓일 위치를 결정한다. 즉 큰 부분으로부터 작은 부분으로 설계하는 하향식(top-down)방식을 취하고 있다. 상향식방식을 취할 경우 절점을 생성하기 위하여 도면으로부터 좌표값을 추출하여야 하고, 일일이 요소들을 정의해야 하기 때문에 많은 노력과 시간을 할애하여야 하나 하향식방식은, 상대적으로 공수 절감의 효과를 기대할 수 있다. Fig.1에서 보듯이 평면(plane)들의 조합으로 구조물의 외형적인 형상 정의를 하며 절점과 요소들은 평면들의 교차에 의하여 자동적으로 구할 수 있다. 3개의 평면이 서로 교차할 때 그 교차점(Fig.3)으로 절점(node)이 생성되며, 2개의 평면이 서로 교차할 때 그 교차선(Fig.2 and Fig.4)으로 요소(element)를 생성하는 하향식방식이 사용되어졌다[11].

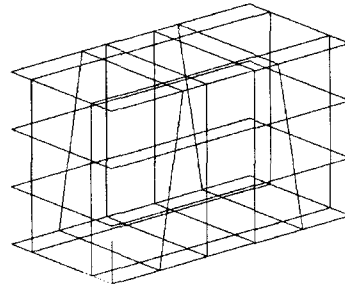


Fig.1 Modelling planes

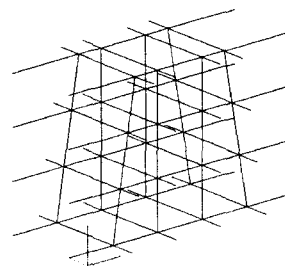


Fig.2 Intersection lines

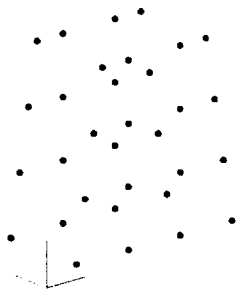


Fig.3 Nodes

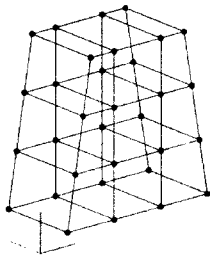


Fig.4 Elements

5. 데이터구조

여러 가지 정보를 컴퓨터 내부에 구축하기 위해서는 데이터를 구조화하여야 한다[12~14].

연결리스트를 바탕으로 하여 다각형을 구현하였다. class Node는 순환 이중 연결리스트로서 데이터 멤버로 자신의 앞을 가리키는 *_prev와 자신의 다음을 가리키는 *_next로 된 두 개의 address를 가진다. 그리고 class Point는 x, y, z의 좌표값만을 가진다.

```
class Node{
protected:
    Node *_next;
    Node *_prev;

public:
    ~~~~ 중략 ~~~~
    Node *remove(void);
};
```

```
};

class Point{
public:
    double x,y,z;
public:
    ~~~~중략 ~~~~
    double length(void);
};
```

위의 두 개의 class를 상속받아 class Vertex를 만들어 class Node 및 class Point를 조작할 수 있게 한다.

```
class Vertex:public Node, public Point{
public:
    Vertex();
    Point(ident,x,y,z){}
    ~~~~~ 중략 ~~~~~
    friend class Polygon;
};
```

class Vertex는 class Point에게 x, y, z 값을 할당하며, class Node에는 address를 할당한다. 또한 friend class로 class Polygon을 정의함으로 class Vertex의 모든 기능들을 사용할 수 있게 하였다.

```
class Polygon{
private:
    Vertex *_OutVertex;
    ~~~~~ 중략 ~~~~~
    int get_angle_num();
};
```

class Polygon은 class Vertex들을 이중연결리스트(Doubly linked list)가 되도록 class Vertex의 class Node를 조작하며, x, y, z의 값은 class Vertex에게 x, y, z의 값을 넘겨준 다음, class Vertex가 class Node에게 정보를 넘겨주게 하였다.

6. 원형요소(Primitive) 정의 및 표현

일반적인 고정식해양구조물(Fixed Offshore Structure)은 대부분 원통, 보 등과 같은 단순한 형상으로 이루어져 있다.

원통형상(Cylinder, tubular pipe)은 양끝 단의 두 개의 반지름(radius1, radius2)과 그 사이 길이(length)로 정의할 수 있다. 또한 보(beam)는 폭(width), 높이(height)와 길이로 정의할 수 있다.

위와 같은 단순한 요소를 원형요소(Primitive)라 부른다.

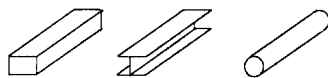


Fig.5 Primitives

이런 원형요소들을 각각의 함수(function)로 정의 한다. 이때 보(beam) 등은 정확하게 정의할 수 있지만 정원(Circle)을 컴퓨터에 표현하는 것이 불가능하므로 정원과 정다각형의 면적비가 99% 이상되는 다각형을 정원으로 가정하여 사용하였다.

위에서 정의한 원형요소 정보를 저장하기 위한 레코드(record)의 필드(field)는 다음과 같이 형상 정보와 위치정보를 갖는다.

형상정보

Type : 형상종류

- 1 : 원통(Cylinder, tubular pipe)
- 2 : L형 보(L-Beam)
- 3 : 사각형 보(Rectangular-Beam)
- 4 : I형 보(I-Beam)
- 5 : T형 보(T-Beam)

Dimension : 형상을 결정하기 위한 치수

	dimension 1	dimension 2
tubular:	시작점의지름(Dia1)	끝점의 지름(Dia2)
L-Beam:	폭(width)	높이(height)
Rect-Beam:	폭(width)	높이(height)
I-Beam:	폭(width)	높이(height)

T-Beam: 폭(width) 높이(height)

위치정보

위치를 나타내는 포인터(Pointer)

7. 초고층강제어초(VEHSAR)의 설계지 원전용시스템 프로그램

7.1. 좌표계

좌표계는 아래와 같이 설정하였다.

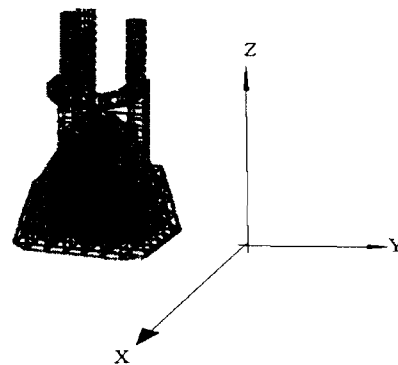


Fig.6 Coordinates system

7.2. 부분 구조 분할

상부, 중앙부, 하부의 기능과 재료의 형상과 규격이 서로 다르고, 구조물의 전체 형상이 복잡하기 때문에 재설계(Redesign)를 쉽게 할 수 있기 위해서 Fig. 7과 같이 상부구조, 중앙부구조, 하부구조로 나눈다.

7.3. 원형요소(Primitive) 정보

원형요소의 형상정보를 사용자가 직접 넣을 수 있는 파라메트릭 설계(Parametric Design) 기능을 구현하였다.

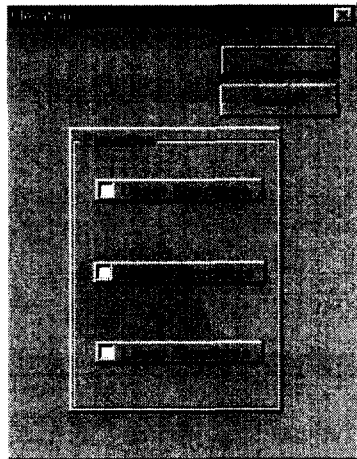


Fig.7 Selection of a elevation

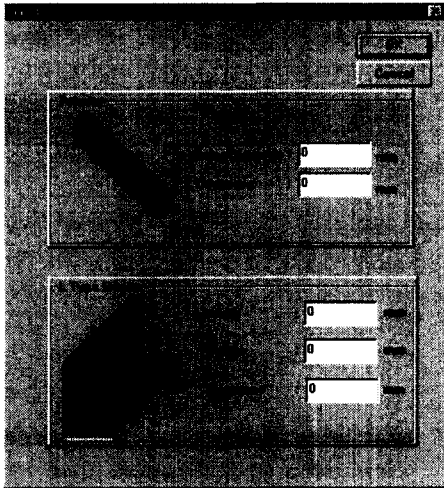


Fig.8 Input of geometric Information on primitives

7.4. 하향식 기법(Top-down Technique)

● Planes (Fig. 9)

도면의 좌표를 결정하여 Polygon들을 배치한다.

● Lines (Fig. 10)

Polygon들이 만나는 intersection line들을 찾아내서 그린다.

● Nodes (Fig. 11)

line과 line이 만나는 점을 찾아내서 그린다.

● Elements (Fig. 12)

Linked List를 이용하여 Node와 Node사이에 선을 하나하나씩 연결한다. 이 때 연결된 선은 방향과 Identity, Names, Material, Member Type을 정의한다.

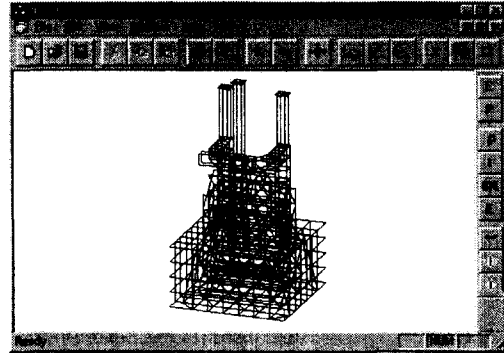


Fig.9 Modelling planes

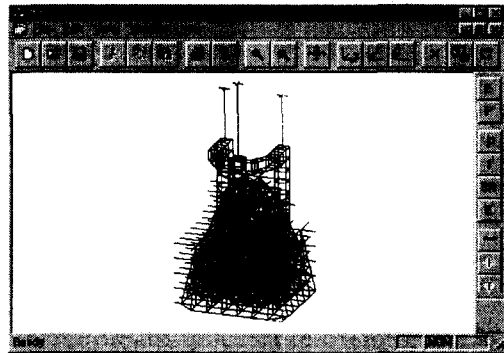


Fig.10 Intersection lines

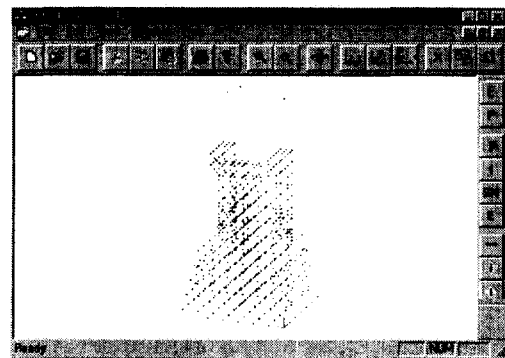


Fig.11 Nodes

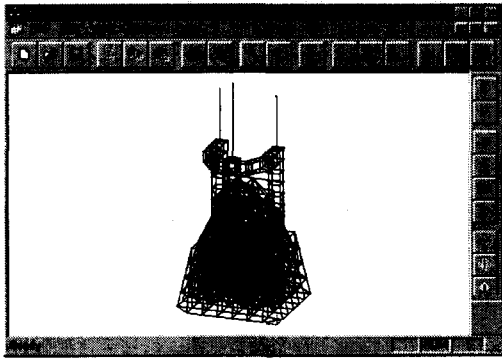


Fig.12 Elements

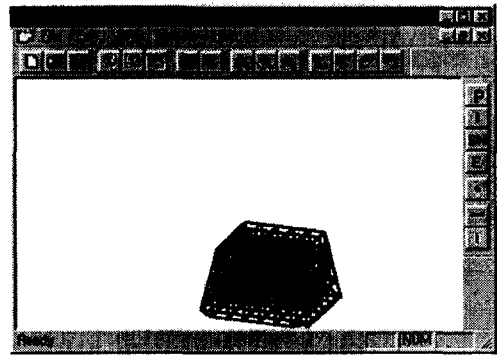


Fig.15 Lower structure

7.5. 부분 구조 생성

각 부분구조를 생성시킨 결과는 다음과 같다.

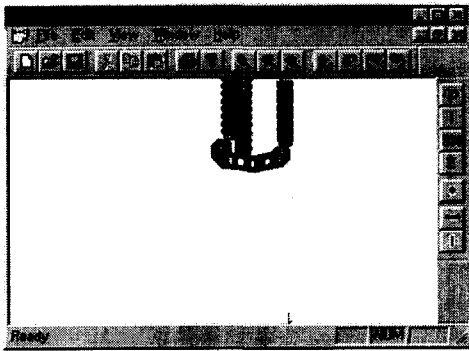


Fig.13 Upper structure

7.6. Solid Modelling

Primitive의 형상정보와 위치정보를 이용하여 각 부분구조의 부재를 통합 배치한 그림은 다음과 같다.

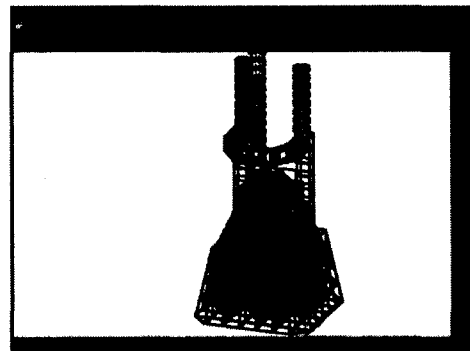


Fig.16 Solid modelling

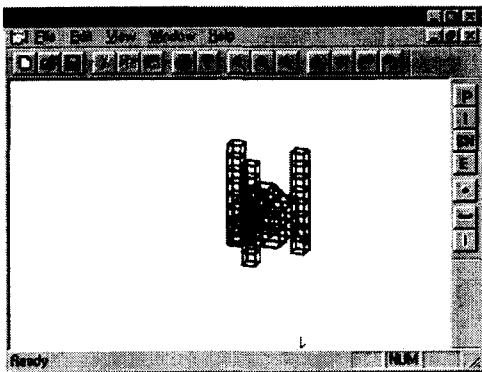


Fig.14 Middle structure

일부분을 확대한 그림은 다음과 같다.

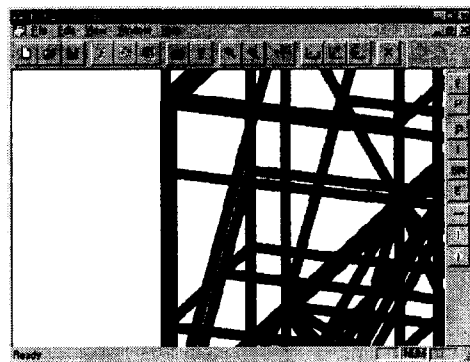


Fig.17 Parially zoom I

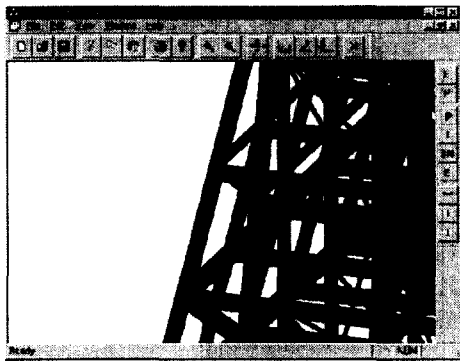


Fig.18 Partially zoom II

7.7. 자재 물량 정보

다음과 같이 각 Primitive의 Member의 고유 번호(number), 형상(type), 재질(material), 치수가 적힌 물량정보를 구할 수 있다.

단위 : mm

===== 물량 정보 =====

Member Number	Member type	Material	Width(Outer Diameter)	Height(Inner Diameter)	Thickness	Length
0	L0	S0	100.00000	100.00000	10.00000	2460.00000
1	L0	S0	100.00000	100.00000	10.00000	2460.00000
===== 중력 =====						
190	L1	S0	100.00000	100.00000	10.00000	2600.00000
191	L1	S0	100.00000	100.00000	10.00000	2600.00000
===== 중력 =====						
200	L2	S0	100.00000	100.00000	10.00000	2250.00000
201	L2	S0	100.00000	100.00000	10.00000	2250.00000
202	L2	S0	100.00000	100.00000	10.00000	2250.00000
===== 중력 =====						
094	C0	A0	32.00000	0.00000	0.00000	3333.78163
097	C0	A0	32.00000	0.00000	0.00000	3333.78163
===== 중력 =====						
1152	L3	S0	75.00000	75.00000	9.00000	1500.00000
1153	L3	S0	75.00000	75.00000	9.00000	1500.00000
===== 중력 =====						
1206	L4	S0	100.00000	100.00000	10.00000	1687.50000
1207	L4	S0	100.00000	100.00000	10.00000	1687.50000
===== 중력 =====						
1202	C2	A0	1500.00000	0.00000	0.00000	350.00000
1203	C2	A0	1500.00000	0.00000	0.00000	350.00000
===== 중력 =====						
1250	C1	A0	32.00000	0.00000	0.00000	550.00000
1251	C1	A0	32.00000	0.00000	0.00000	550.00000

Member number는 Member의 고유 번호이고, Member type은 길이와 단면의 형상에 따라서 나누었다. 단면의 형상이 L형인 경우는 L0~L4까지, 원형인 경우는 C0~C2까지 이다. 그리고 재질(material)은 S0인 경우는 강재(Steel)이고, A0인 경우는 알루미늄(Aluminum)을 나타낸 것이다.

위 정보에 의하여 다음과 같은 자재 물량을 알 수 있다. L0~C2는 길이와 형상에 따라 나눈 Member Type을 나타낸 것이고, L, W, H, t, O.Dia, I.Dia, A, num은 각각 부재의 길이, 폭, 높이, 두께, 외경,

내경, 결면적, 개수를 나타낸 것이다.

단위 : mm

총계

L0 : L*폭*높이*개수 = 2460.00*100.00*100.00*10.00 = 987800.00*10

L1 : L*폭*높이*개수 = 2600.00*100.00*100.00*10.00 = 1043800.00*10

L2 : L*폭*높이*개수 = 2250.00*100.00*100.00*10.00 = 903800.00*696

L3 : L*폭*높이*개수 = 1500.00*75.00*75.00*9.00 = 452530.00*54

L4 : L*폭*높이*개수 = 1687.50*100.00*100.00*10.00 = 678800.00*96

C0 : L*O.Dia*I.Dia*개수 = 3333.78*32.00*0.00 = 673513.23*456

C1 : L*O.Dia*I.Dia*개수 = 550.00*32.00*0.00 = 113801.05*25

C2 : L*O.Dia*I.Dia*개수 = 350.00*1500.00*0.00 = 10367255.76*25

7.8. 재설계(Redesign)

전체 구조물의 길이를 사용자가 직접 변화시켜 구조물을 다시 모델링 할 수 있는 재설계 기능을 구현하였다.

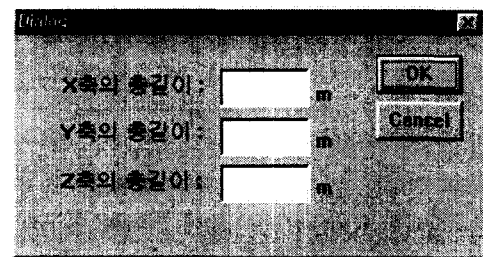


Fig.19 Input of total lengths

구조물의 길이를 변화시켜 모델링을 한 결과는 다음과 같다.

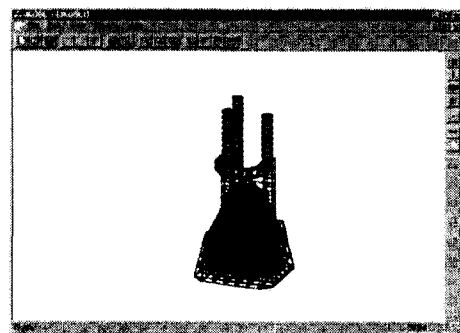


Fig. 20 Solid modeling

8. 결 언

본 연구에서는 3차원 형상정보와 위상정보를 이용하여 Solid Modelling을 할 수 있는 초고층 강재어초(Very High-rise Steel Artificial Reef, VE HSAR) 전용 모델링 프로그램을 개발하였고, 초고층 강재어초의 설계와 적용을 워크스테이션이 아닌 일반 PC에서 사용이 가능하도록 Windows95 또는 Windows NT 환경하에서 MFC(Microsoft Foundation Class)를 기반으로한 Visual C++로 개발하였다. 또한 초고층강재어초 뿐만 아니라 다른 형상을 가진 해양구조물도 설계할 수 있도록 개발되었고, 객체지향프로그램 기법과 CAD/CAM기술의 도입으로 생산성 향상이 기대된다. OpenGL이라는 그래픽 라이브러리를 도입하여 좀더 빠른 고화질의 3차원 그래픽을 화면상에 나타나게 할 수 있다. 그리고 사용자가 직접 설계를 할 수 있는 파라메트릭 설계와 재 설계 기능을 추가하여 반복되는 작업과정을 줄이고 설계변형이 가능하도록 하였고, 자재 물량 정보 데이터를 생성할 수 있다.

후 기

본 연구는 1997년도 교육부 학술연구조성비(해양수산과학 KIOS-97-F-15)에 의하여 연구되었음.

참 고 문 헌

- [1]신현경, 박규원, "고정식 해양구조물의 전산기 지 설계시스템 개발에 관한 연구", 한국해양공학회지, 제9권 제1호, 1995
- [2]전병선, "비주얼 C++ 4.0 MFC 윈도우 95 프로그밍", 삼양출판사, 1996
- [3]곽준기, 백정렬, "Visual C++ 5.0", 삼각형, 1997
- [4]신현경, 박규원, "그래픽 사용자 인터페이스 GUI를 도입한 선형설계 기법에 관한 연구", 대한조선학회 춘계연구 논문발표회, 1993
- [5]한순홍, 이순섭, "전산 선형 설계 시스템의 개발을 위한 기준 모델", 대한조선학회 춘계 연구 논문발표회, 1993
- [6]김원돈, 남종호, 김광욱, "비정규 선형정의를 위한 원형요소 인스턴싱의 이용", 대한조선학회 논문집, 제 30권 제 1호, 1993
- [7]신현경, 손호철, 권명준, 박규원, 송재영, 김종현, 강호승, 이정렬, "타형격벽을 갖는 산적화물선 중앙단면부의 요소생성에 관한 연구", 대한조선학회 춘계연구 논문발표회, 1997
- [8]신현경, 손호철, 권명준, 송재영, 김종현, 강호승, 이정렬, 연구진, "이중선각유조선 중앙단면부의 요소생성에 관한 연구", 대한조선학회 추계연구 논문발표회, 1997
- [9]정중필, "OpenGL Superbible", 에프윈, 1997
- [10]Ron Fosner, "OpenGL Programming for Windows 95 and Windows NT", Addison Wesley Developers Press, 1997
- [11]김이두, "Top-down개념에 의한 건축구조물 설계", 한국과학기술원 박사학위논문, 1993
- [12]Bjarne Stroustrup, "The C++ Programming Language", Addison Wesley, 1991
- [13]Ellis Horowitz, Sartaj Sahni & Dinesh Mehta, "Fundamentals of Data Structures in C++", Computer Science Press, 1995
- [14]Adam Drozdek, "Data Structure and Algorithms in C++", PWS Publishing Company, 1995