

범용 3차원 유동해석용 전/후처리 장치의 개발

허성범*·허남건**

Development of a Pre/Post Processor for a General CFD Code

S. B. Hur*, N. Hur**

Key Words : Pre/Post Processor(전/후처리장치), Mesh Generation(격자 생성), Fire Simulation(화재 모사), Vector & Contour Plot, OpenGL

Abstract

In the present study a pre/post-processor program has been developed to be used with a general CFD code. This program is capable of performing the basic functions of the pre/post-processing, which include mesh generation and post processing plots. Also through perspective projection, this program can be used to check the quality of generated mesh by moving around inside the mesh. The smoke visualization can be also performed with the present program to visualize the smoke behavior in the case of fire simulation. The examples of the program execution are given in paper.

1. 서론

CFD 해석 프로그램은 일반적으로 격자를 생성하는 전처리장치(pre-processor), 생성된 격자를 사용하여 유동을 해석하는 해석 모듈(solver), 그리고 계산결과를 나타내는 후처리장치(post-processor)로 구성된다. 그 중 전후처리장치는 해석을 하기 위한 격자 및 경계조건, 계산조건의 생성과, 해석된 계산결과를 효율적으로 분석하거나 가시화 하기 위한 것으로서, 숫자의 나열로 이루어진 계산 격자나 경계조건, 계산결과를 시각적으로 분석하기 용이하게 해준다. 현재 전산유체역학 분야에서는 PROSTAR^[1], GAMBIT^[2], HyperMesh^[3], ICFM CFD^[4] 등의 전처리 프로그램과 Tecplot^[5], Ensignt^[6], AVS^[7] 등 다양한 후처리 프로그램들이 사용되고 있고, 또한 여러 상용 CFD코드들은 자체적인 전후처리 기능을 포함하고 있는 경우가 많다. 국내에서는 아직 상용화된 CFD 프로그램이 없기 때문에 전후처리 프로그램 또한 개발되고 있지 않으나, 몇 개의 대학에서 기본적인 전후처리기능을 수행하는 프로그램을 개발하고 있다.^[8-9]

본 연구실에서는 기본적인 전후처리 기능과 함께 건물이나 터널 내부에서의 화재 발생 시 연기의 농도와 사람의 시각능력과의 관계를 알아보고, 계산된 결과를 이용 내부의 연기 농도를 3차원 투명도를 사용하여 시각화하는 기능이 포함된 후처리장치를 개발 중에 있다. 과거의 화재 Simulation 연구는 연기의 농도를 그래프형태로 나타내어 실제 화재 시 건물 내부에 있는 사람이 느끼는 시각 상태를 나타내는 데에는 한계가 있기 때문에 3차원적으로 직접 연기를 모사함으로써 직관적으로 연기 분포를 확인할 수 있는데 유용하다.

프로그램 개발은 본 연구실에서 CFD Package 프로그램 개발의 일환으로 제작하고 있는 범용 3차원 유동해석 프로그램에서 사용할 간단한 형태의 전/후처리 프로그램으로 개발하고 있다.

2. Pre/ Post Processor

2.1 개발환경

본 프로그램의 개발환경은 OS Windows 2000 Professional, 1.4GHz P-IV CPU, 512MB RAM으로 개발을 수행하였으며, 사용된 Compiler는 Windows 프로그래밍에 보편적으로 사용되는 Visual C++ 6.0과, MFC(Microsoft Foundation Class Library) 6.0^[10]을 사용하였다. 또한 3D 그래픽스 Library는 OpenGL v1.1^[11]을 사용하였다. 80년대 초반 그래픽스 기술의 필요에 의해 E&S와 Silicon Graphics에 의해 Graphics API로 개발이 시작된 OpenGL은 산업계에서 가장 광범위하게 사용되는 3D Graphics API로 Unix 상의 X-Window 및 Windows NT, Windows 98 등에서 사용이 가능하다. 또한 2D Graphic과는 달리 3차원 그래픽이 표준인 OpenGL은 RealTime Rendering, Shading, Texture Mapping 등 다양한 기능을 가지고 있으며 물체의 표현방법은 3차원 좌표값을 기준으로 하고 있다.

2.2 프로그램 구성

본 프로그램을 실행하면 Fig. 1과 같이 간단한 프로그램 로고가 나타난 후 작업할 Directory를 선택하는 대화상자가 나타난다.

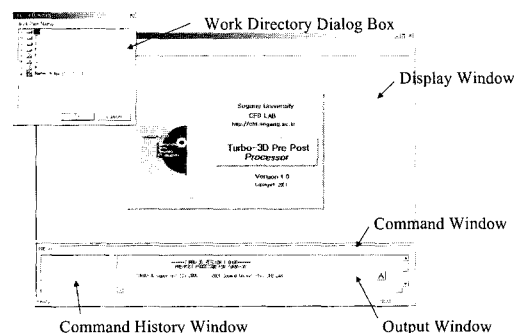


Fig. 1 Window structure of Pre/Post Processor

* 서강대학교 대학원, corscorpjii@kornet.net

** 서강대학교 기계공학과, nhur@ccs.sogang.ac.kr

작업 Directory를 선택하면 메인 윈도우가 나타난다. 메인 윈도우는 크게 4개로 나누어져 있는데 3차원 그래픽 처리를 위한 Display Window와 명령을 입력받는 Command Window, 사용된 명령을 저장하는 History Window, 상태를 출력하는 Output Window로 구성되어 있다. 기본적으로는 MFC구조를 사용하고 있으며, Display Window는 CView class를 상속받아 OpenGL을 사용할 수 있도록 수정하였다.

Command Window로부터 입력받은 명령은 다음과 같은 형식으로 입력을 받는다.

Command, parameter1, parameter2, ...

입력받은 명령은 ' '나 ';'으로 데이터를 구분하여 문자열, 정수, 실수, 테이블에 각각 저장하게 된다. 이때 문자열인 경우에는 앞에서 4글자만을 인식하게 된다. 이렇게 인식된 명령은 명령어 번역기(interpreter)에 의해 실제 명령이 실행되게 된다.

본 프로그램은 Fig. 2와 같은 몇 개의 Command 모듈로 구성되어 있다. 격자점 모듈(Vertex Module)은 격자점을 생성하기 위한 좌표계의 생성 및 변경과 격자점 생성 Command를 포함하고, Cell 모듈(Cell Module)은 격자를 생성하는 Command를 포함한다. 또한 Plot 모듈은 시점의 변경과 화면에 그리는 Command를 포함하는데, 이것은 다시 Geometry, Vector 및 Contour Plot으로 나누어진다. 후처리 모듈(Post Module)은 계산된 데이터를 불러오는 Command와 화재 Simulation을 위한 Command를 포함한다.

2.3 Pre Processor

본 프로그램은 전처리 프로그램의 기능^[2]으로서 다양한 좌표계에서의 격자점과 Cell의 생성, 3차원 그래픽 처리 및 축소, 확대, 회전등의 기능 등을 가지고 있다. Cell 생성방법은 먼저 격자점을 필요한 개수와 위치에 생성한 후, Cell인 경우 8개 또는 Shell인 경우 4개의 격자점이 하나의 Cell이 되도록 Cell을 생성하는 방법이며 조직격자계의 가장 기본이 되는 형태로 제작되었다. 격자점은 현재 선택된 좌표계의 좌표값을 가지게 되며, 실제 저장은 절대 좌표계로 저장되게 된다. 하나의 Cell은 4개 또는 8개의 격자점과 Cell의 형태, Material 번호를 가진다.

Cell이나 격자점의 형태를 관찰하기 위한 view로서 직교투영과 투시 투영의 2가지의 view를 사용한다. 직교투영의 경우 원근감이 없는 투영방법으로 시선 방향으로부터 길이가 항상 일정

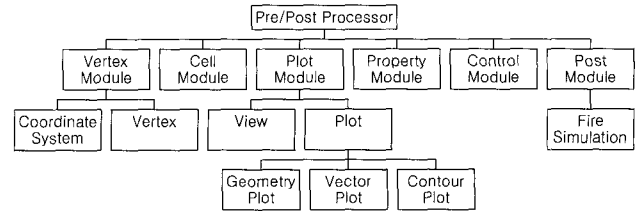


Fig. 2 Command Modules of Pre/Post Processor

한 투영방법이다. 이 투영에서는 회전과 확대/축소를 사용할 수 있다. 본 프로그램은 비조직격자계로의 확장을 고려하여 linked list data 구조를 사용하고 있다. 투시투영은 원근감을 주는 투영 방법으로 관측시점으로부터 멀리 떨어진 물체를 작게 축소된 형태로 나타낸다. 투시 투영에서는 먼저 시점의 위치, 시선의 방향, Up 벡터로 구성되는 시선좌표계를 구성하여 투영하게 되며 키보드의 입력을 통해 시점의 위치를 실시간으로 이동시키면서 내부 격자의 형태와 질을 확인할 수 있다. 이 방법으로 좀 더 실제적인 형태의 격자형태 및 격자의 질을 관찰할 수 있다. 수치 해석에서 격자의 형태와 질은 계산결과에 중요한 영향을 미치는 변수로 작용하기 때문에 격자의 형태와 질을 확인하는 것은 중요하다.

2.4 Post Processor

후처리 기능의 가장 중요하고 기본적인 기능은 실제 계산된 결과를 시각적인 이미지로 표현하여 좀 더 직관적으로 결과를 확인할 수 있도록 하는 것이다. 본 프로그램은 계산된 벡터 및 스칼라 데이터를 이용하여 Vector Plot 및 Contour Plot을 그릴 수 있으며, 명령어 및 마우스를 이용하여 확대/축소 및 회전과 이동을 통해 다양한 View에서 결과의 확인작업이 가능하다. 계산결과는 현재 본 연구실에서 개발하고 있는 해석모듈의 계산결과와 함께, 상용 프로그램인 Star-CD의 계산결과를 사용할 수 있다.

속도 데이터에 대한 후처리 기능으로 Vector Plot을 나타낼 수 있다. 각 Cell이 가지고 있는 속도 데이터를 면의 중심에서부터 스케일의 색과 속도 크기의 화살표로 나타낸다. 스케일은 최대 20개까지 표현할 수 있으며, 전체 Cell과 현재 선택된 Cell중에 최대값과, 최소값을 계산하여 나타낼 수 있으며, 사용자가 임의로 최대값과 최소값을 입력하여 나타낼 수 있다. 화살표의 크기

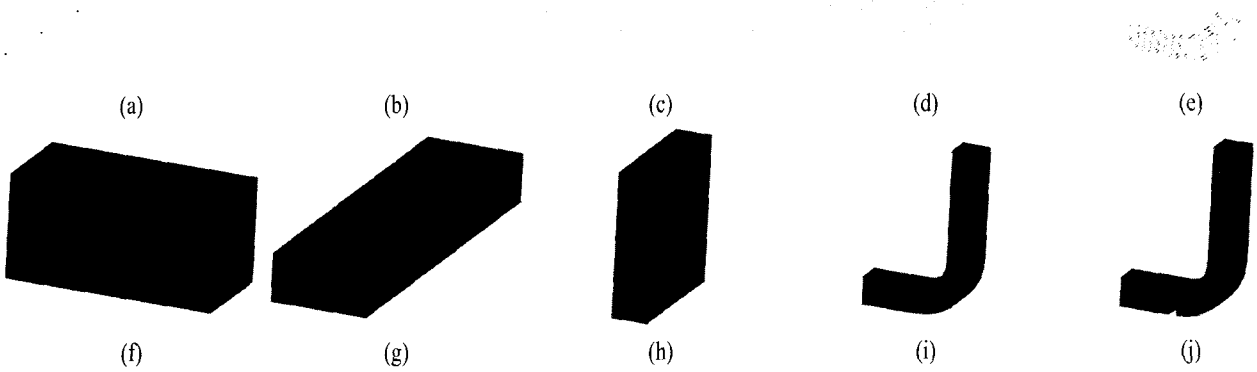


Fig. 3 Example of Mesh Generation

를 변경할 수 있도록 구현되어 있어서 원하는 크기의 화살표로서 속도 등을 표현할 수 있다.

Contour Plot은 스칼라값을 색으로 표현하여 나타내는 방법이다. 기본적으로 본 연구실에서 개발중인 해석모듈은 기본적인 결과 데이터가 모두 Cell 데이터로서 Cell 하나의 색이 일정한 형태이다. 이 경우 Contour의 띠가 부드럽지 않다. 따라서 Cell 데이터를 격자점 데이터로 변환시켜 부드러운 색의 분포로 나타낼 수 있다. 역시 Vector Plot과 마찬가지로 다양한 스케일로 표현이 가능하다.

또한 Line Plot으로 Contour Plot을 나타낼 수 있다. Line Contour Plot의 경우는 한 면을 구성하는 각각의 격자점의 스칼라 값을 이용하여 스케일 값을 찾는 방법을 사용하기 때문에 Cell 데이터로는 구현할 수 없고, 반드시 격자점 데이터로 변환하여야 한다. 각 면의 Contour Line을 찾기 위하여 Marching Squares^[13] 방법을 사용하였다. 이 방법은 각면의 모서리를 구성하는 두 격자점의 스칼라 값을 스케일 값과 비교하여, 스케일 값에 해당하는 Contour Line이 면의 내부를 통과하는 점을 찾아내는 방법이다.

3. 적용예

3.1 격자 생성

먼저 프로그램을 테스트하기 위해 Star-CD Tutorial에 있는 Partially Blocked Elbow 격자를 생성하여 Star-CD와 비교하였다. 격자 생성과정은 Fig. 3에 나타내었다. (a)~(e)는 격자점 생성 과정이며, (f)~(g)는 격자 생성과정이다. (a)에서 v와 vfill명령을 사용해서 그림과 같이 격자점을 만들고, vgenerate 명령을 이용하여 위쪽으로 복사한 후, (c)와 같은 육면체의 형상을 만든다. 그 후 local명령으로 elbow의 회전 중심을 중심으로 하는 원통형 좌표계를 만든 후, csys명령을 이용하여 좌표시스템을 변경한 후 (d)와 같은 형상을 만들고 다시 csys명령을 사용하여 최종적으로 (e)와 같은 형상을 만든다. 그 후 (f)에서 c(cell)명령을 이용하여 Cell 하나를 만들고 cgenerate명령을 이용하여 (g)~(i)와 같은 형상을 만들었다. 최종적으로 cdelete명령을 이용하여 block부분의 Cell을 삭제하여 최종적으로 (j)와 같은 형상을 만들 수 있다.

3.2 격자의 형태 검사

격자의 형태를 검사하기 위한 투시 투영 성능 검증을 위하여 본 연구실에서 수행한 프로젝트인 지하철^[14] 격자를 이용하여 테

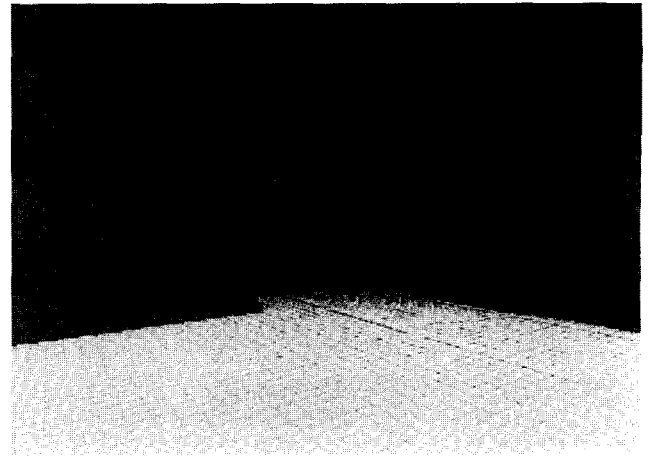


Fig. 4 Perspective View for grid test(Subway station)

스트 하여, 그 결과를 Fig. 4에 나타내었다. Fig. 4는 시점의 위치를 전체 격자의 좌측에 위치시키고, 시선을 우측으로 회전시킨 상태에서 투시투영으로 바라본 그림이다. 역사 내부의 층계와 스크린 도어, 스크린 도어 너머로 문이 열린 지하철의 형상과 지하철 내부의 의자 배치등을 잘 나타내고 있다. 실제 프로그램에서는 키보드를 이용하여 시점을 이동시키면서 지하의 내외부 격자의 형태를 관찰할 수 있다.

3.3 Vector & Contour Plot

3.1에서 생성한 격자를 이용하여 본 연구실에서 개발중인 해석모듈을 이용하여 계산한 후 그 결과를 이용하여 후처리 작업을 하여, Fig. 5에 나타내었다. (a)는 유동속도에 대한 Vector Plot이다. 유동의 방향에 따른 화살표의 방향과 크기가 잘 나타나고 있다. (b)는 Cell 데이터를 이용한 압력의 Contour Plot이다. Cell 데이터를 이용하여 표현하였기 때문에 Cell 하나가 같은 색을 갖는 것을 볼 수 있다. 따라서 연속적인 Contour로 나타나지 않고 불연속적인 모양을 보인다. (c)는 격자점 데이터를 이용하여 나타낸 압력 Contour Plot이다. 이 경우에는 격자점과 격자점 사이를 부드럽게 연결하여 표현하기 때문에 (b)에서와 같이 불연속적이지 않고, Cell 내에서도 부드러운 색의 분포로 나타난다. (d)는 Contour Line으로 압력의 Contour를 나타낸 그림이다. Cell 내부의 압력 Contour가 연속된 선으로 부드럽게 연결되어 있음을 볼 수 있다.

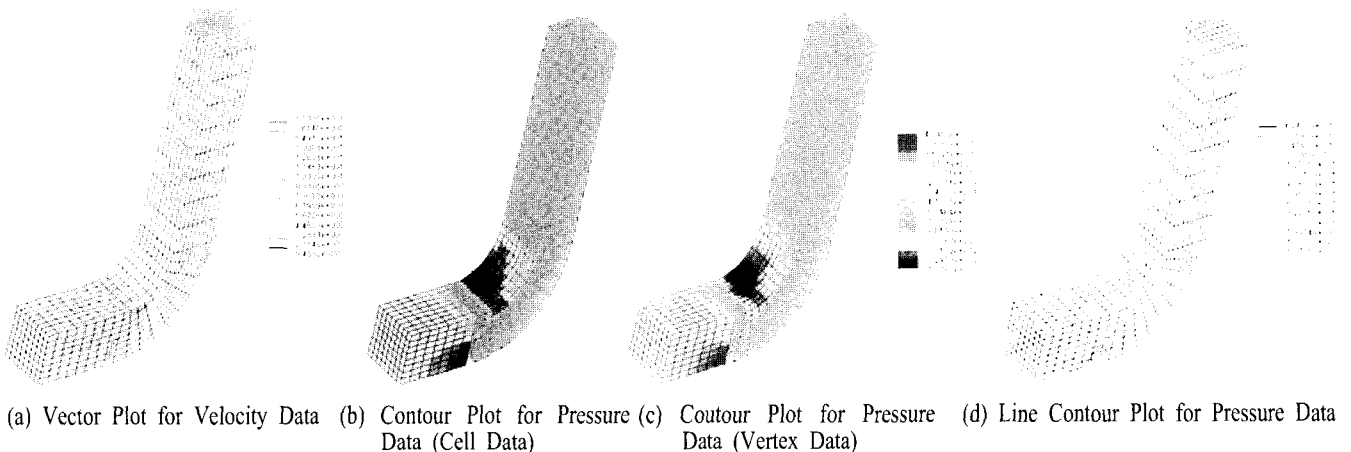


Fig. 5 Post Plot Type

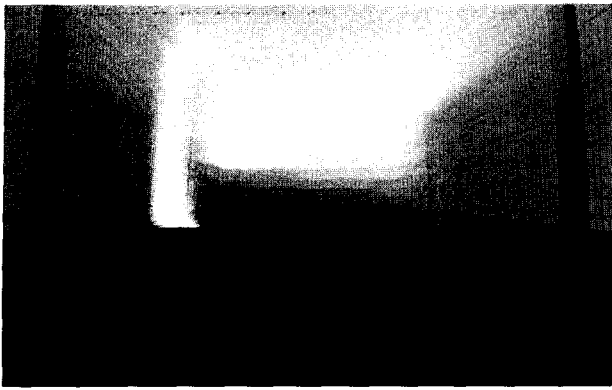


Fig. 6 Smoke concentration distribution inside train

3.4 화재 Simulation

본 연구실에서 수행한 프로젝트인 경량전철시스템의 화재 시뮬레이션의 결과^[15]를 이용하여 본 프로그램의 Blending 기능을 이용한 화재 Simulation의 성능을 테스트 하였다.

Ray Tracing을 이용하여 시선방향의 연기의 농도를 적분하는 방법^[8]은 렌더링 질은 우수하지만, Cell이 많아질수록 계산량이 많아지게 된다. 따라서 시점을 이동시킬 경우에 렌더링 시간이 길어지게 된다. 이같은 속도를 개선하기 위해 OpenGL내의 Blending^[16]기능을 이용하여, 시선방향에 농도를 적분하지 않고, 직접 거리에 따른 연기농도값을 투명도로 갖는 유체Cell을 그려 줌으로써, 적분하는 방식보다 렌더링 시간을 감소시켰다. Fig. 6은 화재발생시 60초가 지난 후의 연기분포를 모사한 것이다. 화재발생지점과 연기가 천정으로 올라간후 다시 밑으로 퍼지는 현상이 잘 모사되어 있다. 그러나 연기가 열은 부분에서 Cell의 겹치는 부분으로 인해 렌더링 질이 낮아지는 것을 볼 수 있다. Fig. 7은 두가지 방식에 대한 렌더링 시간에 대한 그래프이다. Step이 증가할 수록 열차의 중심부에서 앞으로 시선이 이동된다. 농도를 적분하는 방법의 경우 시선의 위치에 따라 렌더링 시간의 변화가 크고 평균적으로 더욱 많은 시간이 걸리는 반면에 Blending을 사용한 경우 시점의 움직임에 관계없이 약 3초의 일정한 시간이 소요됨을 알 수 있다.

4. 결론

본 연구에서는 범용 CFD 프로그램 개발의 일환으로 CFD 해석을 위한 전/후처리 프로그램을 개발하고 다양한 경우에 대하여 적용하였다. 본 프로그램은 계산에 필요한 격자를 다양한 좌표계를 통하여 생성하는 기능 뿐만 아니라, 투영 좌표계를 이용하여 생성된 격자 내부의 형태를 실시간으로 시점을 이동시키면서 관찰할 수 있다. 또한 계산된 결과를 이용하여, Vector Plot과 Contour Plot등의 후처리 Plot을 표현할 수 있으며, 마우스를 이용하여 확대, 축소, 이동, 회전등을 할 수 있어, 다양한 View와 Size로 격자 및 계산 결과를 관찰할 수 있다. 또한 화재 simulation에서 발생한 연기의 분포를 3차원 투시 투영을 이용하여 가시화 할 수 있다. 이 프로그램은 향후 다양한 기능 및 GUI 기능들을 추가하여 본 연구실에서 개발하고 있는 범용 3차원 유동해석 프로그램의 전용 전/후처리 장치로 개발을 진행할 계획이다.

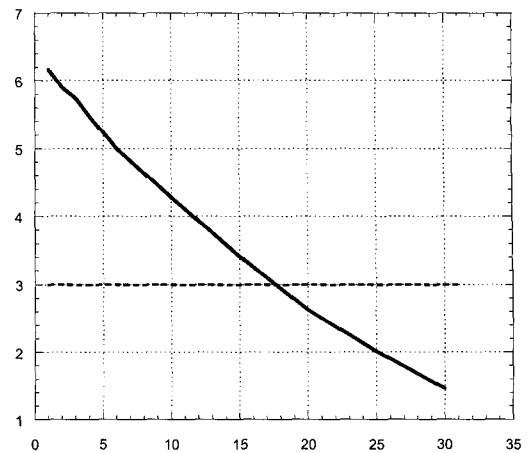


Fig. 7 Comparison of rendering time for smoke visualization

후기

본 연구는 과학기술부의 공학용 해석 소프트웨어 기술개발사업의 연구비지원(M1-012900001)으로 수행되었으며, 이에 관계자 여러분께 감사드립니다.

참고문헌

- [1] <http://www.cd.co.kr>
- [2] <http://www.fluent.com>
- [3] <http://www.altair.com>
- [4] <http://www.icemcfd.com>
- [5] <http://www.amtec.com>
- [6] <http://kw-tech.co.kr>
- [7] <http://www.neat.co.kr>
- [8] 장재원, 허남건, “화재 Simulation을 위한 Post Processor 개발,” 한국전산유체공학회 2001년도 춘계학술대회논문집(2001), pp.155-160.
- [9] 나정수, 김기영, 김병수, “입체 구현 기능을 지닌 데이터 분석 및 가시화 프로그램의 개발,” 한국전산유체공학회 2002년도 춘계학술대회논문집(2002), pp.158-163.
- [10] 이상엽, Visual C++ Programming Bible, 영진(2002).
- [11] <http://www.opengl.org>
- [12] 장재원, 김사량, 허남건, “CFD 해석을 위한 3차원 격자생성 프로그램의 개발,” 한국전산유체공학회 2001년도 추계학술대회논문집(2001), pp.157-163.
- [13] Edward Angel, Interactive Computer Graphics, Addison-Wesley(2000).
- [14] 허남건, 서울 지하철 9호선 1구간 건설공사 승강장 선로부 TES 환기 Simulation 및 화재 배연 Simulation, 화승(2001).
- [15] 허남건, “경량전철 시스템 안전프로그램 계획을 위한 화재 방재 대책수립,” 경량전철시스템 기술개발사업 2차년도 연구결과보고서, 한국철도기술연구원(2000).
- [16] Richard S. Wright, Jr. Michael Sweet, OpenGL SuperBible, Waite Group(2000).