

# 교육 훈련용 3차원 항공기 시뮬레이터의 구현\*

최성윤\*\*, 채상원\*\*, 한영신\*\*, 이철기\*\*\*

## 3D Flight Simulator for Education of Flying Tactics

sung-yun Choi, sang-won Chae, young-sin Han, chil gee Lee

### Abstract

The flight simulator should be made like a actual flight. For the scene of sight, instrument should show the condition of flight and the pilot should catch the altitude, speed, pose and rate of lift of the airplane.

The paper describes 3D flight visual training program of driving airplane in practice. It is for beginners using joystick in PC, implements airplane physical equations. And it uses rendering technology to implement vision parts of flying object.

\* 본 논문은 한국 시뮬레이션 학회 2003년 춘계학술대회에서 발표한 내용을 수정, 보완한 것임.

\*\* 성균관대학교 정보통신공학부 컴퓨터공학전공

\*\*\* 성균관대학교 정보통신공학부 교수

## 1. 서론

### 1.1 연구 배경

시뮬레이터는 시뮬레이션의 대상이 되는 원래의 시스템과 비슷하게 동작하도록 제작되어 사용자에게 실제 시스템을 이용하는 것과 같은 효과를 주는 것으로, 연구용, 훈련용, 오락용 등으로 그 사용 목적이 다양하다.

이러한 시뮬레이터 중에서 비행 시뮬레이터는 현재 국내 각 항공사 및 비행 훈련소, 군에서 사용되고 있다. 1996년을 기준으로 전 세계 시뮬레이터 시장은 약 200억불에 달하며 이중 150억불 가량이 군용으로 추산된다. 이러한 시장 규모는 군비 감축의 세계적 흐름에도 불구하고 계속 증대하고 있는데, 특히 동아시아 시장의 성장세는 괄목할 만하다. 민간 항공기의 조종사 훈련용 시뮬레이터 시장은 1980년대 말에 정점을 이루었으나 1990년대에 접어들면서 항공 산업의 침체와 더불어 급격히 감소하였고 1996년 이 규모의 3분의 1 하로 감소하는 불황을 경험하였다. 그 결과 거대 시뮬레이터 기업들이 도산을 하였으며 이러한 기술 인력의 유출이 새로운 시뮬레이션 시장의 시발점으로 이어졌다. 이와 더불어 시뮬레이션 산업은 다양성과 규모 면에서 성장을 하였고 향후 엄청난 시장 규모가 예측되고 있다[4][6].

국내 시장의 경우 국방 부문에서 일찍이 미국으로부터 장비와 기술을 이전받아 비교적 낮은 등급의 모의 훈련장치로부터 최첨단, 최고급 시뮬레이터에 이르기까지 운용 차원에서의 시뮬레이션 경험은 가히 국제적인 수준이다. 또한 민간 차원에서는 대한 항공과 아시아나 항공이 각기 운용과 정비 유지에서 세계적 수준에 이르렀다. 그러나 이에 반해 국내외 시뮬레이션 관련 연구, 개발 수준은 선진국과 많은 차이를 보이며 대부분의 기술을 해외에 의존하고 있는 형편이다. 더욱이 국방과 관련된 시뮬레이션 기술의 경우는 보안과 관련, 더욱더 해외에 미치지 못하는 형편이다.

### 1.2 연구 목적

비행 시뮬레이터는 컴퓨터가 장족의 발전을 하기 전에도 부분적인 시뮬레이터가 사용되었지만 지금은 시뮬레이터 탑승 시간의 일부를 항공 법규상 비행시간으로 인정하는 수준까지 시뮬레이터가 발전하였다. 국내 민간항공업체에서는 이미 이를 이용한 훈련과정이 체계화 되어 있고 군에서도 시뮬레이터를 구입하여 훈련에 활용하기 시작하였다.

미 공군의 한 보고서에서 보면 시뮬레이터를 이용하여 훈련할 때보다 상당한 예산 절감이 된다는 것을 볼 수 있다. 이는 비행의 무경험자를 실제 비행과 비슷한 상황에서 비행을 경험하고 비행 훈련을 하면 그만큼 훈련 시간이 줄어든다는 것을 나타낸다. 아울러 악천후, 계기 고장 등 비정상 상태에 대한 경험은 그보다도 더욱 더 효과적 가치를 갖고 있다.

이런 수준의 시뮬레이터를 구현하는 데 있어서 비행 시뮬레이션은 과거 10년 동안 복잡한 계산의 증가에 따라 시뮬레이션 호스트 컴퓨터의 계산량도 점점 증가하게 되었다. 이로 인해 컴퓨터의 실시간 처리를 위해 여러 개의 프로세서를 사용하여 각기 다른 일을 처리하게 하는 방법을 사용하였다.

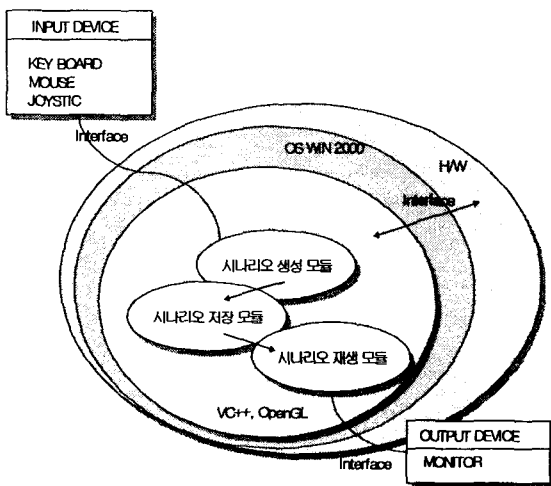
이 때문에 시스템의 가격 상승을 불러왔고 재정적으로 어려운 곳에서는 프로세서 하나 또는 두 개 정도를 사용하여야 했기 때문에 실시간 처리가 불가능하였다. 한 두 개의 적은 프로세서나 컴퓨터에서 실시간 처리를 위해서는 최대한 연산 처리 부분을 줄이는 방법을 사용하여야 하는데 모델을 최소화 시켜도 프로세서 자체 큰 부담을 가지게 된다. 이를 위해서 모델의 최소화와 비선형 부분은 항공기 제원으로부터 계산된 Look-Up table의 계수를 사용하여 처리함으로써 계산량을 줄이는 방법을 사용하였다. 또한 실시간 처리를 가능하도록 하여 조종간의 조작으로부터 신호를 테이블에서 검색하여 그에 맞는 데이터를 모션 방정식에 의해 계산하여 비행 자세, 위치 등을

계산해 주게 된다. 이를 위해 실시간 시뮬레이션에 대한 기본적인 개념과 그를 만족시킬 수 있는 비행 시뮬레이터의 기본적인 수학적 모델에 비행 역학적인 데이터와 조종간의 탄성 방정식을 이용하여 최종적인 비행 자세와 위치에 대한 데이터를 얻게 된다.

일반적으로 항공기 시뮬레이터는 크게 시뮬레이션 엔진, 지형정보 DB, 렌더링 엔진으로 구성되어 있는데 본 논문에서는 이 중에서 시뮬레이션 엔진과 렌더링 엔진을 중점으로 항공기 시뮬레이션에 대하여 연구하였다. 구현된 항공기 시뮬레이터를 이용하여 비행 조종 훈련사들과 실제 비행 조종사들에게 공중 전투 기동, 요격, 저고도 침투 및 공격 등과 같이 다양한 환경에서 3차원을 통해 이루어지는 비행 훈련을 지상학술 단계에서 조종사들이 보다 쉽게 이해하도록 전달하는 것을 주목적으로 한다.

## 2. 시스템 모델링

본 논문에서 구현된 시스템은 전술 기동 훈련에 대한 교육 목적으로 항공 전술 훈련에 대한 시나리오를 저작하며, 저작된 시나리오를 3차원 그래픽으로 표현하여 교육의 전달 및 습득할 수 있는 시스템이다. <그림 1>은 구현된 시스템이 작동하는 구성도를 보여준다.

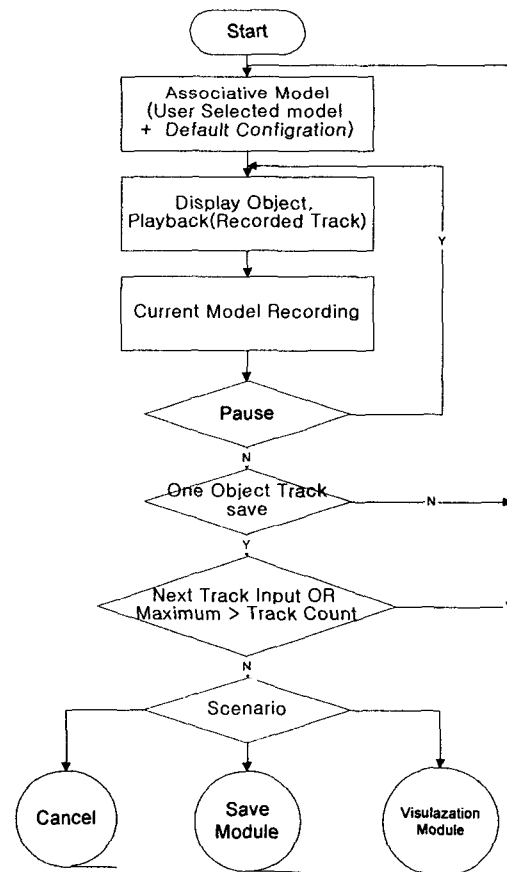


<그림 1> 시스템 구성도

본 논문에서 구현한 시스템은 키보드, 마우스와 조이스틱 등의 입력 장치(Input Device)를 통해 개체 이동경로를 입력하고, 입력된 이동 경로 좌표대로 시스템 내부의 개체(전투기 등)를 이동시켜 시나리오를 생성한다. 또한 생성된 시나리오를 3차원으로 재생하며 3차원 공간상의 개체를 사용자가 원하는 위치로의 재생을 제어할 수 있다. 생성된 시나리오는 시나리오 파일로 하드디스크에 저장 할 수 있다.

### 2.1 시나리오 생성 모듈

<그림 2>의 Flowchart는 본 논문에서 구현한 시스템의 시나리오 생성 모듈에 관한 것이다.



<그림 2> 시나리오 생성 모듈

- 1) Associative Model : 완성된 시스템에서 제공하는 환경과 사용자가 원하는 객체 등을 불러온다.
- 2) Display Object, Playback : 1)에 의해 불러진 객체 등을 화면상에 나타내 준다.
- 3) Current Model Recoding : 2)를 통해 불러진 객체 등을 마우스나 조이스틱 등을 이용하여 조종하는 것을 레코딩한다.
- 4) Pause : 레코딩이 멈추면 다음 단계로, 그렇지 않으면 2)의 과정으로 계속하여 반복한다.
- 5) Object Track Save : 1)~3)을 통해서 녹화된 기록을 파일로 저장한다.
- 6) Next Track Input OR Maximum > Track Count : 다른 트랙을 생성하고자 하는 경우에는 1)의 과정으로 돌아간다.
- 7) Scenario : 생성된 파일을 시나리오 별로 작동하게 한다.
- 8) Cancel : 시나리오 생성을 취소한다.
- 9) Save Module : 생성된 시나리오를 저장한다.
- 10) Visualization Module : 생성된 시나리오를 사용자에게 보여준다.

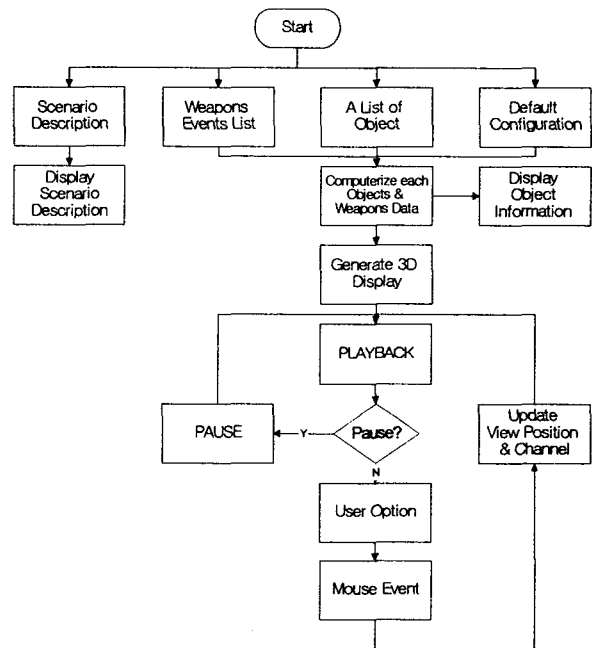
## 2.2 Visualization 모듈

Visualization 모듈은 2.1에서 생성되어지는 시나리오를 사용자가 직접 확인하면서 그 내용을 이해할 수 있게 한다.

- 1) Scenario Description : 2.1을 통하여 생성되는 파일에 기록될 시나리오에 대한 설명을 입력하는 것이다. 각각의 비행 전술이나 상황 등이 기록되어 진다.
- 2) Display Scenario Description : 1)의 과정을 사용자에게 보이도록 한다.
- 3) Weapons Event Scenario : 2.1에서 생성될 파일에 기록될 내용 중 무기에 관련된 내용을 불러온다. 본 논문에서 구현된 시스템에서는 지형을 제외한 비행기나 무기 등은 모두 객체로 구성되어 있으므로 이에 대한 각각의 조종을 통한 시나리오 생성은

필수적이다.

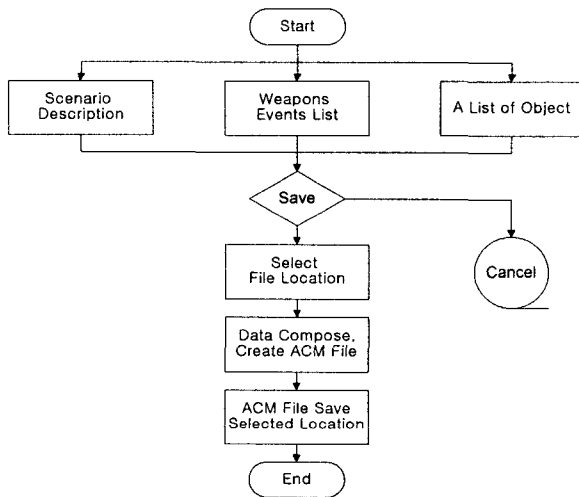
- 4) A List of Object : 2.1에서 생성될 파일에 기록되는 여러 개의 객체의 리스트를 의미한다.
- 5) Computerize each Objects & Weapons Data : 4)에 의해 불러진 리스트를 통하여 각각의 객체와 무기의 데이터를 불러와서 연산한다.
- 6) Generate 3D display : 2.1에서 생성될 파일에 기록되는 내용 모두를 불러와서 사용자에게 보여준다.
- 7) Pause : 사용자에게 출력중인 내용을 멈추거나, 8)~9)의 과정을 진행한다.
- 8) User Option : 사용자는 옵션을 통하여 현재 진행 중인 시나리오를 다양한 뷰 포지션과 채널을 통하여 볼 수가 있다.
- 9) Mouse Event : 마우스 이벤트를 이용하여 현재 진행 중인 시나리오를 다양한 뷰 포지션으로 볼 수가 있다.



<그림 3> Visualization 모듈

### 2.3 시나리오 저장 모듈 (Save Module)

<그림 4>는 2.1에 의해서 생성된 시나리오를 구현된 시스템의 포맷에 맞게 저장하는 것을 보여주는 flowchart이다.



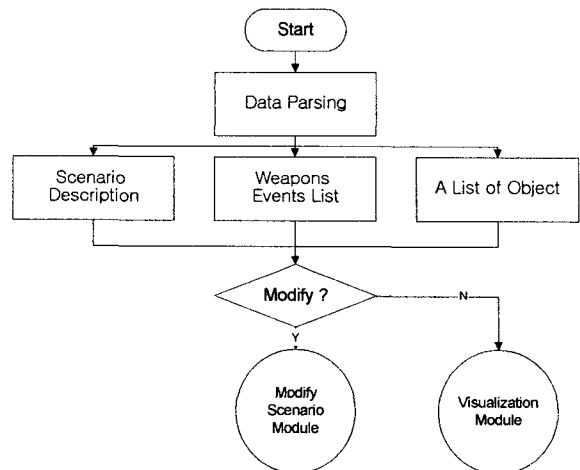
<그림 4> 시나리오 저장 모듈

- 1) Scenario Description : 2.1과 2.2를 통해서 생성되어진 시나리오의 설명을 기록한다.
- 2) Weapons Events List : 1)과 마찬가지로 2.1과 2.2를 통하여 생성되어진 시나리오에 쓰인 무기 리스트를 기록한다.
- 3) A List of Object : 2.1과 2.2를 통하여 생성되어진 시나리오에 쓰인 객체들에 관한 리스트를 기록한다.
- 4) Save : 생성된 시나리오를 저장할지를 결정한다.
- 5) Select File Location : 저장할 위치를 정하도록 한다.
- 6) Data Compose, Create ACM File : 저장 위치가 정해지면 1)~3)의 내용을 모두 혼합하고 ACM이란 형태의 파일을 생성한다.
- 7) ACM File Save Selected Location : ACM 파일을 정해진 위치에 저장한다.

### 2.4 시나리오 재생 모듈

<그림 5>는 2.1과 2.3에 의해서 생성되고 저장된 시나리오를 구현된 시스템에 의해 재생하는 것을 보여준다.

- 1) Data Parsing : 사용자가 원하는 시나리오를 저장된 파일 중에서 골라서 불러온다.
  - 2) Scenario Description : 저장된 파일에 기록된 시나리오의 설명을 불러온다.
  - 3) Weapons Events List : 저장된 파일에 기록된 시나리오에 쓰이는 무기들을 불러온다.
  - 4) A List of Object : 저장된 파일에 기록된 시나리오에 쓰이는 객체들을 불러온다.
- 2)~4)의 과정이 끝난 후 사용자의 선택에 의해 저장된 파일을 볼 수 있고, 혹은 수정할 수 있다.



<그림 5> 시나리오 재생 모듈

## 3. 시스템 환경

본 논문에서 구현된 시스템의 환경은 다음과 같다.

&lt;표 1&gt; 시스템 환경

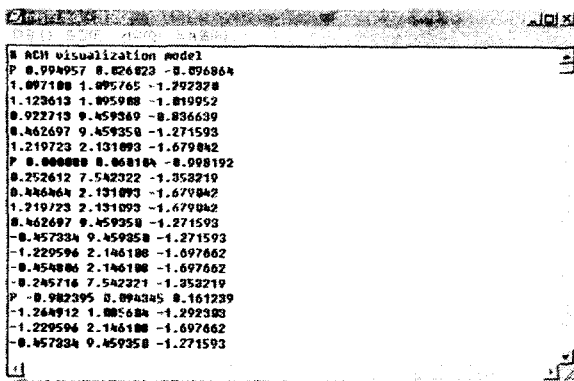
항 목	내 용
OS	MS Windows 2000 Server
CPU	Pentium4 2.0 GHz
RAM	256M
Graphic Card	GeForce4 계열
개발 Tool	MS Visual Studio 6.0 Ent.
3D API	OpenGL

#### 4. 3차원 항공기 시뮬레이터의 구현

항공기는 이륙에서부터 착륙까지의 전 과정을 수행을 하지만 본 논문에서 구현한 항공기 시뮬레이터의 주목적은 비행 조종 훈련사들이나 실제 조종사들이 비행 전술 등을 보다 쉽게 이해하기 위함이 목적이므로 이륙, 착륙을 배제하였다. 비행 상태에서 각각의 객체인 비행기들의 비행을 통해 조종사들이 쉽게 이해할 수 있도록 비행 시뮬레이션 시나리오를 이용하여 구현하였다. 여기에 사용되는 입력 장치로 조이스틱과 마우스 등을 이용하였다.

##### 4.1 시뮬레이션 객체

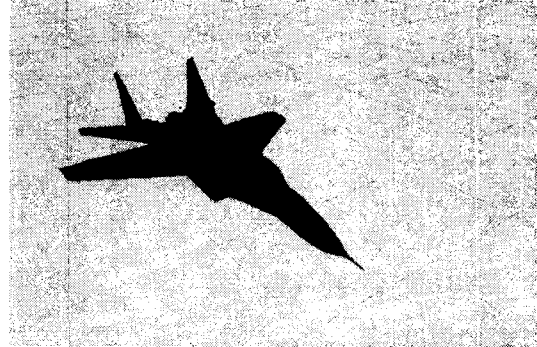
본 논문에서 구현된 항공기 시뮬레이터는 각각의 객체를 하나의 파일로 생성하여 만들 수 있게 하였다. 단순히 항공기 객체만을 생성할 수 있는 것이 아니라 여러 종류 형태의 객체를 생성할 수 있다.



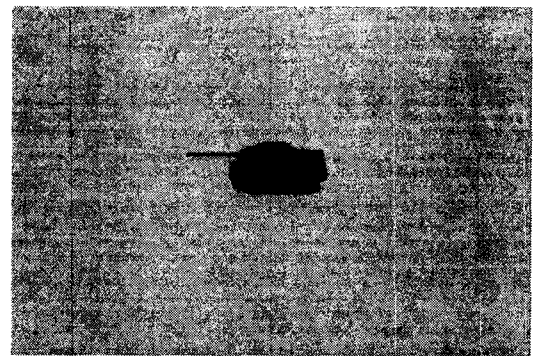
&lt;그림 6&gt; MIG 29 항공기를 모델링한 소스 파일

위 <그림 6>은 본 프로그램에서 구현된 항공기 객체를 모델링한 소스 파일이다.

아래 <그림 7, 8>은 <그림 6>의 과정을 통해서 모델링된 모습이다.



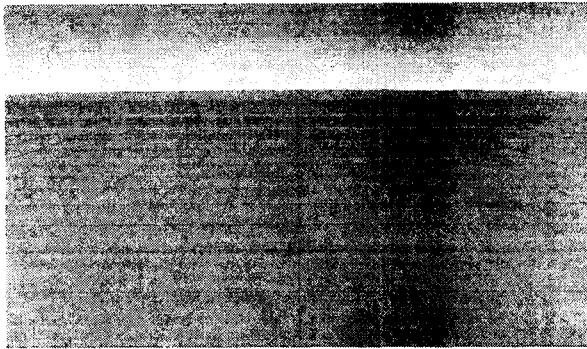
&lt;그림 7&gt; MIG 29 항공기를 모델링한 모습



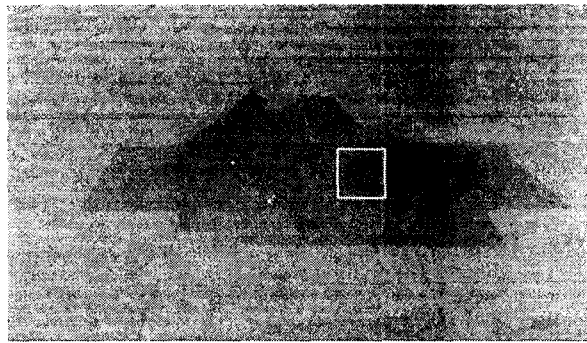
&lt;그림 8&gt; 탱크를 모델링한 모습

##### 4.2 지형

구현된 시뮬레이터에 쓰이는 기본 지형은 <그림 9>과 같으며, 객체를 추가하여 <그림 10>와 같이 기본 지형에 산을 만들 수 있다. 하지만 현재 상태에서는 객체의 수가 제한되어 있기 때문에 일정수 이상의 산과 같은 여러 객체를 추가하여 실제 지형과 비슷하게 만들 수는 없다. 실제의 지형 정보를 입력하기 위해서는 <그림 6>처럼 실제 지형을 파일로 만들어서 객체를 생성할 수가 있다.



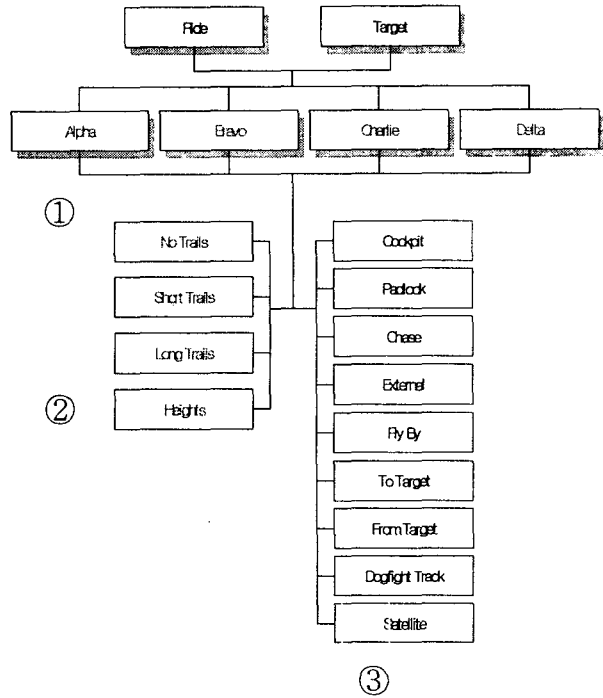
<그림 9> 기본 지형



<그림 10> 기본 지형에 산을 추가한 모습

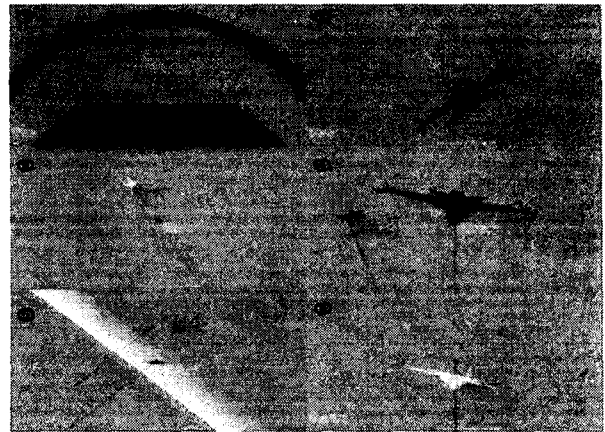
### 4.3 View Channel Control

본 논문에서 구현된 프로그램은 사용자에게 표현된 3차원 그래픽모드의 시야변환을 메뉴 바(Menu Bar)의 Full-Down형태로 하며, 사용자는 메뉴바의 선택, 단축키의 사용으로 시야 변화를 제어하고 사용자 시야 변화의 항목들은 <그림 11>의 구성도의 항목들로 구성한다. ①은 비행 객체를 선택하고, ②는 각 비행 객체의 고도와 위치 정보를 표시하고, ③은 비행 객체를 사용자에게 여러 가지 환경으로 보여 준다.



<그림 11> View Channel Control

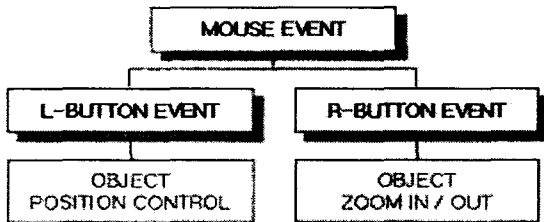
<그림 12>는 View Channel을 변경함으로써 시뮬레이션 중인 화면을 여러 가지 뷰 모드로 볼 수 있게 해주는 예제이다.



<그림 12> View Channel Control

#### 4.4 View Position Control

3차원 공간상에 표현된 시나리오의 각 객체의 위치, 객체와의 거리를 사용자는 마우스 드래그에 의하여 제어 하고, 마우스 이벤트를 처리하여 3차원 공간상의 객체 시뮬레이션 재생모드를 제어한다.

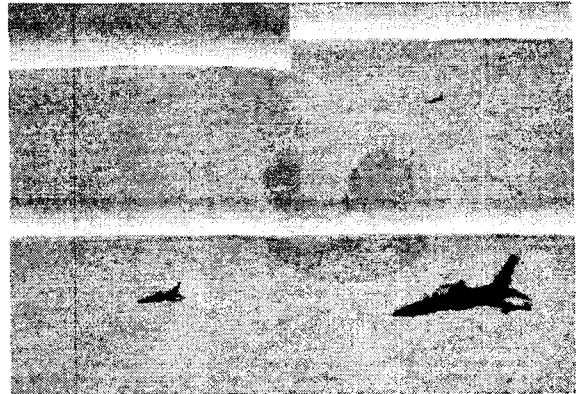


<그림 13> View Position Control

<그림 13>는 마우스 이벤트에 따라서 View Position이 바뀌는 것을 나타낸다. <그림 14>은 마우스의 좌측 버튼을 이용하여 이벤트를 발생하였을 경우 사용자에게 객체의 모습을 여러 가지 각도에서 보이는 화면이며, <그림 15>은 마우스의 오른쪽 버튼을 이용하여 이벤트를 발생할 경우 객체를 Zoom-In, Zoom-Out하여 볼 수 있는 예제 화면이다.



<그림 14> View Position Control using Left Button



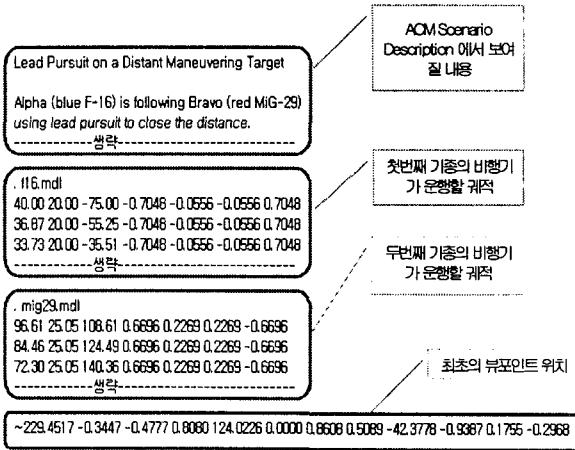
<그림 15> View Position Control using Right Button

#### 4.5 Operation

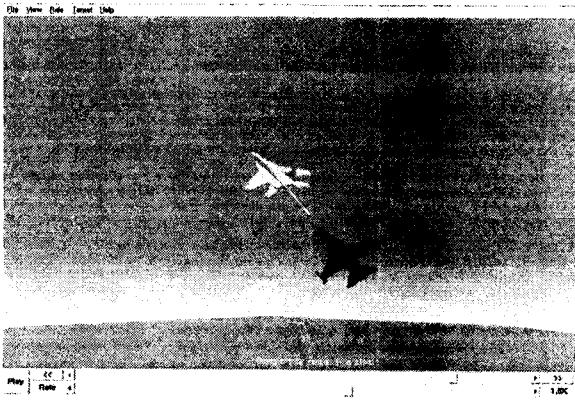
본 논문에서 구현된 항공기 시뮬레이터는 우선 객체를 생성하고, 그 객체를 기본 지형에 불러온 후 입력 장치를 이용하여 각각의 객체를 조종하게 되어 있다. 그 객체는 <그림 6>과 같이 이루어져 있으며, 그 객체들 간의 비행 시뮬레이션은 다음 파일의 형식으로 저장 이 된다. <그림 16>은 저장된 파일의 예제 이 다.

<그림 17>는 구현된 항공기 시뮬레이터의 인터페이스이다. 메뉴바 형식으로 되어 있으며 <그림 11>과 같은 기능이 제공된다. <그림 18>은 항공기 시뮬레이터를 작동하는 예제 이 다. <그림 19>은 각각의 항공기 객체가 비행 시뮬레이션을 하는 동안의 비행 정보가 사용 자에게 보여지는 예제이다. 이를 통해서 사용 자는 자신의 기체가 어떤 비행을 하고 있는 지 를 손쉽게 파악할 수 있으며, 저장된 시나리오 를 통해 여러 사용자들이 비행 시나리오를 보 다 쉽게 이해, 숙지하기가 가능해진다.

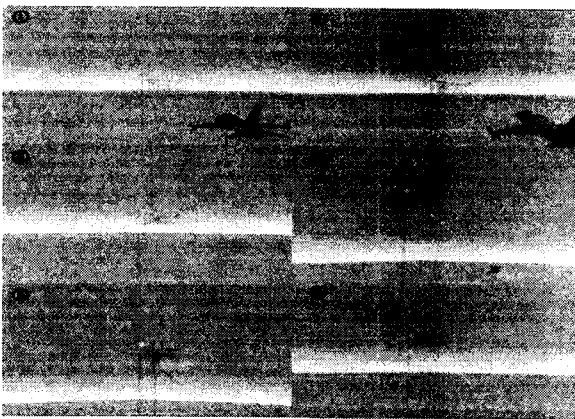




<그림 16> 작성된 시뮬레이션을 저장하는 파일 형식



<그림 17> 항공기 시뮬레이터의 인터페이스



<그림 18> 항공기 시뮬레이터 작동 예



<그림 19> 각각의 항공기 객체에 대한 비행 정보

### 5. 결론 및 추후 연구

본 논문에서는 비행 조종 훈련사와 실제 조종사들을 위한 항공기 시뮬레이터를 구현하였다. 조종사들이 3차원으로 제작된 비행 시나리오를 참고하면서 그에 해당하는 내용을 숙지할 수 있도록 하며, 실제 시뮬레이터를 조작하여 자신이 원하는 비행을 작성하고, 파일로 기록하여 추후 자신의 모의 비행을 다시 볼 수도 있다.

이러한 목적으로 구현된 항공기 시뮬레이터이지만, 아직 미비한 점이 몇 가지 있다. 우선 실제 지형을 정확히 구현하지 못한 점이다. 현재는 조종사들에게 비행 기술에 관한 교육 목적을 기본으로 하였기 때문에, 기본 지형을 토대로 그 위에 객체를 덧붙임으로 인해서 지형을 나타내는데 크게 문제는 없으나 보다 정확하고 정밀한 시뮬레이터가 되기 위해선 입체의 지형을 지형 정보를 이용하여 사용자에게 시각화 하는 것이 필요할 것이다.

## 참고문헌

- [1] J. M. Rolfe, K. J. Staples, "Flight Simulation", Cambridge aerospace series, 1997
- [2] 한국항공우주연구소, "Development of a R&D Flight Simulator(4)", 과학기술처, 1989
- [3] Barnes, A. G., "Modelling requirement in flight simulation", Aeronautical Journal, vol.98, pp 395-404, 1994
- [4] 윤석준, "항공기 시뮬레이터 기술의 현재", 전자공학회지, 제25권 제2호, pp172-181, 1998
- [5] 한국 항공 우주 진흥 협회, "비행시뮬레이터 개발", 항공우주, pp18-21, 1994.11
- [6] 윤석준, "모의훈련 장비 구현을 위한 모델링 및 실시간 시뮬레이션 기법 연구", 한국 항공 우주학회지, 27권 4호, pp143-149, 1999
- [7] Mille, Gavin S. P., "The Defining and Rendering of Terrian Maps", Proceedings of ACM SIGGRAPH, vol. 20, no. 4, August 1986
- [8] Martin Reddy, Yvan Leclerc, Lee Iverson, Nat Bletter, "TerraVision II: Visualizing Massive Terrian Database in VRML", IEEE Computer Graphics and Applications, Vol. 19, No. 2, 1999
- [9] Allan Watt, Mark Watt, "Advanced Animation and Rendering Techniques", Addison-Wesley, 1992
- [10] Andrea L. Ames, David R. Nadeau, John L. Moreland, "VRML 2.0 Sourcebook, Second Edition", Wiley, 1996
- [11] Richard S. Wright JR., Michael Sweet, "OpenGL Superbible", Waite Group Press, 1996
- [12] John De Goes, "C++ 3D Game Programming, Cutting-Edge", Coriolis Group Books, 1997
- [13] Averill M. Law and W. D. Kelton, "Simulation Modeling & Analysis", McGraw-Hill, 1991

주 작성자 : 최 성 윤

논문투고일 : 2003. 9. 10

논문심사일 : 2003. 10. 22

심사판정일 : 2003. 10. 22

## ● 저자소개 ●



## 최성운

2002 부경대학교 공과대학 제어계측공학과 학사  
 2002 성균관대학교 공과대학 컴퓨터공학과 석사4기  
 관심분야: 컴퓨터 모델링 및 시뮬레이션, 3D Scanner



## 채상원

2001 성균관대학교 공과대학 제어계측공학과 학사  
 2003 성균관대학교 공과대학 컴퓨터공학과 석사  
 관심분야: 컴퓨터 모델링 및 시뮬레이션, 생체인식



## 한영신

1997 이화여자대학교 대학원 전산정보 공학석사  
 2003 성균관대학교 컴퓨터공학과 박사과정  
 1994~1996 (주) Simtech 연구원  
 현재 인천시립대학교 컴퓨터공학과 강사  
 관심분야 : 컴퓨터시뮬레이션, 공장자동화, 패턴인식



## 이칠기

1980 성균관대학교 전자공학과 졸업  
 1979~1983 한국방송공사 근무  
 1985 Arizona State University 전기 및 컴퓨터 공학과 석사  
 1990 University of Arizona 전기 및 컴퓨터 공학과 박사  
 1990~1995 삼성전자 수석연구원  
 1995~현재 성균관대학교 전기전자 및 컴퓨터 공학부 부교수  
 관심분야 : 컴퓨터 시뮬레이션, 객체지향 모델링, 전문가 시스템