

# 건설 차량 실시간 그래픽 주행 시뮬레이터

손권\*, 최경현\*\*, 유창훈\*\*\*

## A Real-Time Graphic Driving Simulator of the Construction Vehicle

Kwon Son\*, Kyung Hyun Choi\*\*, Chang Houn You\*\*\*

### ABSTRACT

A graphic software is one of the most important components of the vehicle simulator. To increase a visual reality of the simulator, the graphic software should require several technologies such as three-dimensional graphics, graphic modeling of the vehicle and the environment, drivers biomechanical models, and real-time data processing.

This study presents a real time graphic driving simulator of a construction vehicle. The graphic simulator contains the three models of the construction vehicle, the human, and the environment, and employes a neural network approach to decrease an on-line dynamic computation. An excavator model is represented using an object-oriented paradigm and contains the detailed information about a real-size vehicle. The human model is introduced for objective visual evaluations of the developed excavator model. Since the environment model plays an important role in a real-time simulator, a block-based approach is implemented and a text format is utilized for easier construction of environment. The simulation results are illustrated in order to demonstrate the applicability of developed models and the neural network approach.

**Key Words:** Driving Simulator(주행 시뮬레이터), Construction Vehicle Model(건설차량 모델), Human Model(인체 모델), Environment Model(환경 모델)

### 1. 서론

각종 차량과 중장비, 항공기 분야의 설계 및 제작 단계에서 시뮬레이터의 역할이 크게 부각되고 있다. 시뮬레이터를 이용하면 제작 기간 및 비용을 절감할 수 있을 뿐만 아니라 실제 차량을 제작하여 실제로 재현하기 힘든 상황을 가상적으로 재현할 수 있으므로 차량의 테스트 등에 유용하게 사용할 수 있다. 이미 선진국에서는 운전 시뮬레이

터를 이용하여 제한된 상황에서 운전 및 작업 상황을 재현하여 운전자가 운전 감각 및 작업 활동을 익힐 수 있도록 하고 있다.<sup>(1-4)</sup>

시뮬레이터에서 운전자는 주로 시각을 통해 주행 상황과 환경을 인지하므로 실시간 그래픽 처리는 현실감 재현 면에서 큰 역할을 담당한다. 아이오와 주행 시뮬레이터<sup>(5,6)</sup>는 운전석 전·후면의 화면에 각각 수평시야 190°와 65°의 고해상도의 텍스처 그래픽을 투영시켜 사용하였다. 주행 시나리오

\* 부산대학교 기계공학부 및 기계기술연구소

\*\* 제주대학교 기계에너지생산공학부

\*\*\* 부산대학교 기계공학과 대학원

## 2. 시뮬레이터의 구성

를 작성하기 위하여 타일 형식(tile-based)을 기초로 한 정사각형의 영역을 이용하여 차량 주행 환경을 제작하였다. 벤츠자동차회사에서는 1984년부터 운전 시뮬레이터를 제작하여 차량의 모의실험에 이용하고 있으며, 지속적인 연구를 수행하여 1994년에는 시각처리를 담당하는 영상투사등을 개선하여 진보된 형태의 시뮬레이터를 완성하였다.<sup>(7, 8)</sup> 국내에서도 자동차 주행공간의 바닥을 타일로 구분하여 운전 연습기에 응용하여 사용한 연구가 발표되었으며,<sup>(9)</sup> OpenGL 그래픽 라이브러리를 이용하여 PC에서 그래픽 처리를 수행한 시뮬레이터가 제작하였다고 보고하였다.<sup>(10)</sup>

그러나, 굴삭기와 같은 건설차량을 기존의 차량 시뮬레이터에 적용하기 위해서는 추가적인 사항이 요구된다. 굴삭기는 일반적인 주행 위주의 차량과는 달리 비교적 저속의 주행속도로 운행되지만, 작업장치의 영향이나 작업영역을 고려한 시야 평가가 중요하다. 이러한 정량적인 평가를 위해서는 주로 인체 모델이 사용되며, 중장비의 설계에 도움을 주기 위하여 인체모델을 이용하여 시야평가를 할 수 있는 프로그램<sup>(11)</sup>도 개발되었다. 차량 시뮬레이터에서 차량모델은 바퀴의 회전등 미소한 부분을 제외하면 고정된 3차원의 자료만으로 충분히 구성할 수 있다. 중장비의 경우는 주행뿐 아니라 작업 상황도 고려해야 하므로 중장비의 각 작업장치가 적절한 움직임이 가능하도록 모델링하여야 한다.

본 연구에서는 일반 차량용 시뮬레이터를 확장하여 건설차량인 굴삭기 주행 시뮬레이터에 응용할 수 있는 실시간 그래픽 시뮬레이터를 개발하였다. 개발된 그래픽 시뮬레이터는 많은 계산 시간을 요구하는 온라인 동역학 해석 대신으로 실시간 처리를 위해 해석 결과의 학습을 구현하는 신경망 접근법을 사용하였으며, 많은 데이터와 계산량을 요구하는 주행환경의 그래픽 처리 속도를 향상시켜 운전자에게 높은 현실감을 주는 방법론을 도입하였다. 운전자뿐 아니라 시뮬레이터 사용자에게도 그래픽 시뮬레이터에 쉽게 접근할 수 있도록 객체 지향접근법을 통한 시스템을 모델링하였으며, 인체 모델을 통해 시야범위 등 정량적인 평가에 의해 구성된 주행 환경을 생성하고, 작업 상황을 재현할 수 있는 굴삭기 그래픽 모델을 개발하였다.

굴삭기 시뮬레이터는 Fig. 1과 같이 6자유도 운동재현기와 운전석, 유압 시스템, 전용 제어기, 워크스테이션, 그래픽 이미지 처리장치 등으로 구성된다. 운동 재현기는 실제 굴삭기의 운동을 그대로 재현할 수 있도록 한정된 범위 내에서 유압시스템에 의해 3방향의 병진 운동과 3축 중심의 회전 운동이 가능하도록 되어 있다. 운동 재현기의 위 부분에 놓이게 될 운전석에는 실제 굴삭기와 동일한 조작장치가 장착되어 있으며 이 조작장치의 조작신호를 센서를 이용해 측정할 수 있도록 되어 있다. 전용 제어기는 80c196 마이크로프로세서로 구성되어 운동 시스템의 자세 제어와 워크스테이션간의 통신을 담당한다. 영상 처리장치는 워크스테이션에서 만들어진 이미지를 프로젝터를 통해 운전석 앞에 위치한 스크린에 영상을 올려준다. 워크스테이션은 그래픽 이미지를 생성해 주는 역할과 전체 시스템을 관리하는 역할을 한다. 이러한 그래픽 소프트웨어는 운전 상황을 시각적으로 처리함으로써 굴삭기 시뮬레이터의 핵심 요소이다.

시뮬레이터에서 요구되는 실시간 그래픽 처리는 운전자의 조작이 입력되면 동역학 해석을 빠르게 수행하여 결과를 받아서 그래픽처리를 빠르게 처리하여 실제 조작에 의한 움직임과 화면에 재생되는 그래픽 화면이 일치되는 것을 말한다. 실제 굴삭기 플랫폼에서 구현에 앞서 이러한 그래픽 시뮬레이션의 실시간 처리와 완벽한 굴삭기, 인체, 주변 환경 모델의 그래픽 생성의 검증이 요구된다.

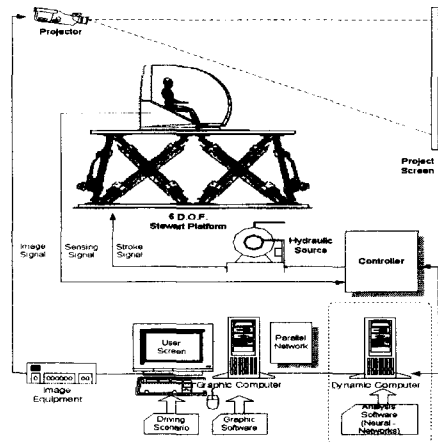


Fig. 1 Excavator driving simulator

### 3. 그래픽 모델

굴삭기 주행 그래픽 시뮬레이터에 사용되는 그래픽 모델은 건설 차량인 굴삭기 모델, 인체 모델, 주행 환경 모델로 크게 분류된다. 굴삭기 모델은 객체 모델링 기법(object modelling technique)을 도입하여 실제 제품화된 굴삭기를 기준으로 개발하였으며, 인체 모델은 외부 응용 프로그램을 이용하여 생성된 데이터를 재구성하여 만들었다. 주행 환경 모델은 텍스트 형태로 제작하여 사용자가 원하는 주행 도로를 일반 텍스트 편집기를 이용하여 쉽게 구성할 수 있도록 하였다. 주행 환경은 데이터 양이 많아 처리 시간이 오래 걸리므로 이를 개선할 수 있는 컬링프로세스(Culling process) 및 상세 레벨(level of detail) 알고리즘 등의 적용이 용이하도록 모델링하였다.

#### 3.1 굴삭기 모델

굴삭기의 객체 모델링은 Fig. 2와 같이 시스템 전체를 객체들로 구성하고 이들 객체들 사이의 관계를 정의한다. 집합관계는 ◇로 표기하고 논리적인 연결관계는 -로 표기한다. 굴삭기는 하부 프레임과 상부 프레임, 작업 장치를 구성하는 붐과 디퍼, 버킷, 그리고 동력 및 구동장치로 구성된다. 굴삭기 특성에 맞추어 각 작업장치는 운동이 가능한 형태로 모델링하고, 실제 굴삭기 치수를 적용하여 재현한다. 주행시에는 작업장치와 상부 프레임은 고정시켜 하나의 물체로 가정하고 동역학 해석이 수행되지만, 작업 상황에서도 사용할 수 있도록

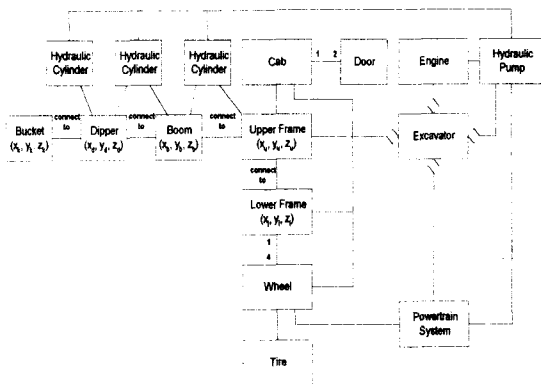


Fig. 2 Excavator object model

작업장치를 모델링한다.

굴삭기의 운동부위를 기술하기 위해 굴삭기를 계층적인 구조로 구성하고 이를 표현하기 위해 계층적 좌표계를 사용한다. 상부 프레임은 하부 프레임에 종속되고, 붐은 상부 프레임에, 그리고 디퍼와 버킷은 붐에 종속시킨다. 상위 좌표계가 움직이면 이에 연결된 하위 좌표계도 같이 움직이므로 한 부분의 움직임에 대한 다른 부분의 변화를 고려하지 않아도 된다. 예를 들면 상부 차체를 회전시키면 상위 좌표계에 속하는 하부 차체는 움직이지 않고, 하위 좌표계에 연결된 캡과 작업장치가 같이 움직인다. 그러므로 상부 차체의 변화에 따른 작업장치의 절대 위치를 다시 계산하지 않아도 된다.

굴삭기의 운동을 기술하기 위해 Fig. 3과 같이 굴삭기의 앞쪽을 -x 방향, 위쪽을 +z 방향이라고 좌표계를 정의하고, y 방향은 오른손법칙에 따라 결정된다. 굴삭기의 주행해석에 필요한 절대 위치와 자세를 나타내는 변환  $T_1^w$  은 다음과 같다.

$$T_1^w = T_{xyz} R_{xa} R_{ya} R_{za} = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \quad (1)$$

$$\begin{bmatrix} C_x C_z & -C_y S_z & S_y & 0 \\ C_z S_x S_y + C_x S_z & C_x C_z - S_x S_y S_z & -C_y S_x & 0 \\ -C_x C_z S_y + S_x S_z & C_z S_x + C_x S_y S_z & C_x C_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases} C_x = \cos \alpha_x, S_x = \sin \alpha_x \\ C_y = \cos \alpha_y, S_y = \sin \alpha_y \\ C_z = \cos \alpha_z, S_z = \sin \alpha_z \end{cases}$$

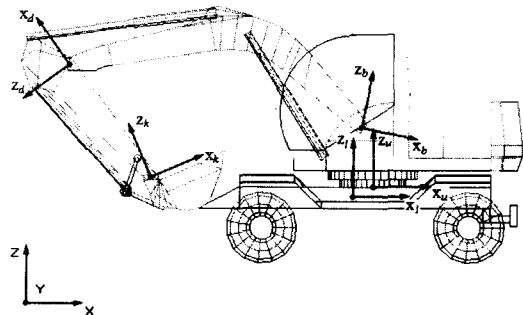


Fig. 3 Coordinates of excavator model

여기서  $T_{xyz}$ 는 굴삭기의 절대 이동 변위를 나타내는 변환이며,  $R_{xa}$ ,  $R_{ya}$ ,  $R_{za}$ 는 굴삭기 질량 중심의 지면에 대한 회전을 나타내는 변환이다. 또한,  $x_1, y_1, z_1$ 는 굴삭기 하부프레임의 절대 위치 좌표값이며,  $\alpha_x, \alpha_y, \alpha_z$ 는 굴삭기의 자세를 나타내는 값으로 각각 x, y, z 축에 대한 회전을 나타낸다.

Fig. 4에서와 같이 굴삭기의 작업에 따르는 좌표계의 변환에서 작업변경을 나타내는데 사용되는 상부프레임의 하부프레임에 대한 z축으로 회전하는 변환  $T'_u$ 은 다음과 같다.

$$T'_u = \begin{bmatrix} 1 & 0 & 0 & x'_2 \\ 0 & 1 & 0 & y'_2 \\ 0 & 0 & 1 & z'_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha_u & -\sin \alpha_u & 0 & 0 \\ \sin \alpha_u & \cos \alpha_u & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

여기서  $\alpha_u$ 는 상부프레임의 회전 각도를 나타내며,  $x'_2, y'_2, z'_2$ 은 하부프레임에서 상부프레임의 회전 중심까지의 거리를 나타내며, 실제 모델링에 사용된 값들은 각각 -634.8, 66.8, -1388.4 mm이다.

Fig. 5에서와 같이 굴삭기의 작업 영역을 나타내는데 사용되는 변환들인  $T_b^a, T_b^b, T_k^d$ 는 작업 장치인 붐, 디퍼, 버킷의 y축의 회전에 의해서 다음과 같이 구해진다.

$$T_b^a = T_{xyz3} R_{ya_b} = \begin{bmatrix} 1 & 0 & 0 & x'_3 \\ 0 & 1 & 0 & y'_3 \\ 0 & 0 & 1 & z'_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha_b & 0 & \sin \alpha_b & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha_b & 0 & \cos \alpha_b & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_d^b = T_{xL_b} R_{ya_d} = \begin{bmatrix} 1 & 0 & 0 & -L_b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha_d & 0 & \sin \alpha_d & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha_d & 0 & \cos \alpha_d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_k^d = T_{xL_d} R_{ya_k} = \begin{bmatrix} 1 & 0 & 0 & -L_d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha_k & 0 & \sin \alpha_k & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha_k & 0 & \cos \alpha_k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

여기서  $L_b$ 과  $L_d$ 는 각각 붐과 디퍼의 길이를 나타내는 상수로 4300 mm와 2000 mm가 사용되었으며,  $\alpha_b$ 는 작업 장치의 회전 각도를 나타내는 변수이고,  $x'_3, y'_3, z'_3$ 은 상부차체에서 굴삭기 붐의 회전 조인트가 부착된 위치까지의 거리로 각각 280.9, 0.0, 337.2 mm가 사용되었다.



Fig. 4 Rotation of upper frame

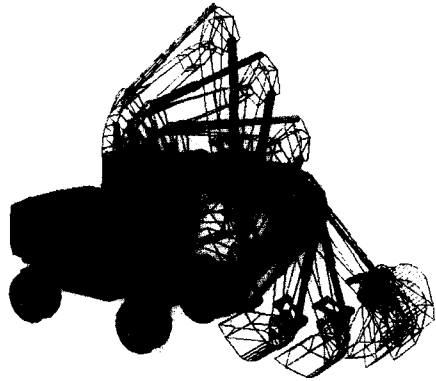


Fig. 5 Movements of attachments

### 3.2 인체 모델

본 연구에서는 시뮬레이터의 객관성 있는 주행 환경의 표현과 굴삭기 내에서의 시야평가를 위하여 생성한 인체 모델을 굴삭기 내부에 위치시켜 분석하였다. 사용된 인체 모델은 미공군에서 개발한 GEBOD<sup>(9)</sup>을 이용하였다. GEBOD은 비행기 조종사의 거동을 해석하는 프로그램에 사용하기 위한 인체 모델의 자료를 생성하는 프로그램이지만, 생성된 인체 모델은 비행기뿐 아니라 승용차, 트럭과 같은 차량의 운전자 모델에도 이용할 수 있다.

GEBOD 프로그램을 실행시키면 확장자가 AIN인 인체 데이터 파일을 제작할 수 있는데, AIN파일은 인체 모델의 동역학 해석 프로그램인 ATB<sup>(10)</sup>에서 직접 사용할 수 있는 형태의 파일이다. 이 파일은 사용자가 직접 이용하기에는 난해한 숫자의 나열로 이루어져 있으나 컴퓨터를 통한 연산에 사용하기에

Table 1 AIN file format

filename : ttt.ain										
17	15	human		ttt		CARD B.1				
LT 1	24.870	92020.84511	0672	4.718	6.942	4.322	-0.472	0.000	0.885	1 CARD B.2
		0.00	0.00	0.0						
CT 2	5.100	14480.08090	2189	4.389	6.092	3.341	-1.463	0.000	0.105	1 CARD B.2
		0.00	0.00	0.0						
LT 3	52.874	35863.21112	7229	4.826	6.451	7.122	0.000	0.000	0.136	1 CARD B.2
		0.00	14.40	0.0						
P G 1	0	-1.45	0.00	-2.24	-2.39	0.00	2.25			CARD B.3
		0.00	0.00	0.00	0.00	5.00	0.00			0 0 0
W H 2	0	-1.76	0.00	-0.83	-0.32	0.00	6.99			CARD B.3
		0.00	0.00	0.00	0.00	5.00	0.00			0 0 0
SP K 3	0	-0.27	0.00	-7.26	-0.93	0.00	1.46			CARD B.3
		0.00	0.00	0.00	0.00	10.00	0.00			0 0 0

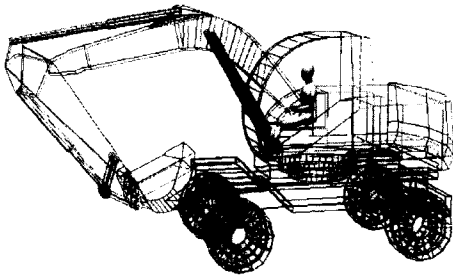
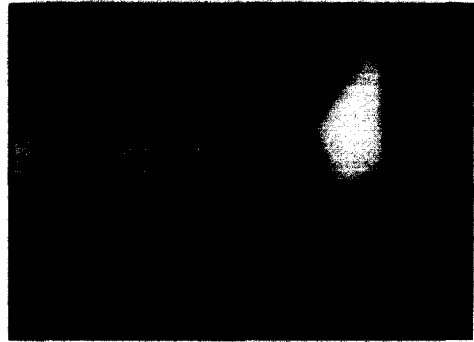


Fig. 6 Human model in vehicle

는 편리하게 구성되어 있다. Table 1은 사용된 인체 모델의 AIN 파일의 내용을 나타낸 것으로 첫 줄의 숫자는 각각 인체모델의 지체와 관절의 개수를 나타낸다. 지체와 관절의 개수를 기준으로 파일에 기록된 값을 차례대로 읽어 내려간다. 각 줄의 끝에 위치한 CARD 번호에서 B.2는 지체를, B.3은 관절에 관련된 정보임을 나타낸다. 본 연구에서는 인체모델의 동역학적 해석은 하지 않고 기하학적 정보만을 이용하므로 CARD 번호 B.4 이상의 물리적 정보는 이용하지 않는다.

인체 모델을 그래픽적으로 표현하기 위해 각 지체는 GEBOD에서 가정한 것과 동일하게 타원체로 타내었다. 각 타원체는 인체의 관절과 지체의 중심 위치를 기초로 계산되어 하나씩 구성된다. 사용된 인체 모델은 엉덩이를 기준으로 하부 몸통, 상부 몸통, 목, 얼굴의 순서로 생성된다. 인체의 지체가 생성되는 것과 같은 순서에 따라서 시점의 위치도 결정된다.

인체 모델을 굴삭기 내부의 의자에 위치시킨 후 상반신과 얼굴의 움직임을 고려하여 주행 환경을 모델링한다. 인체의 시야범위와 시점이 향하는 방향은 지정된 범위 내에서 변경 가능하다. Fig. 6은 인체 모델을 굴삭기 내부에 위치했을 때의 모습이며, Fig. 7은 인체 모델 삽입 전후의 시플래이더



(a) Screen without human model



(b) Screen with human model

Fig. 7 Screens with and without human model

에서 사용될 그래픽 화면을 비교한 것으로, (a)는 인체모델 없이 제작된 화면이고, (b)는 인체 모델을 삽입한 후 자세를 조정하여 시점을 보정한 화면이다.

### 3.3 주행 환경 모델

주행 환경은 굴삭기를 지지하고 동시에 반력이 가해지는 지면과 주변을 구성하는 건물, 가로등 등 주변 환경 요소로 구별 될 수 있다. 주행 환경의 뼈대가 되는 지면은 주로 도로에 의해 구현된다. 도로는 사용자가 원하는 임의의 구조를 표현할 수 있어야 하며, 도로의 구성에 맞게 주변 환경도 조성되어야 한다.

주행 환경은 블록 단위로 구성하였다. 블록개념은 기존의 타일개념을 보완한 것으로 타일이 일정한 크기와 형상의 조각을 하나의 층으로 채우는 반면에 블록은 큰 블록이 또 다른 작은 블록을 포함하여 두 개 이상의 층으로 채워지는 형상으로 구성된다.

본 연구에서는 세 개의 층으로 구성되는 블록 구조를 주행 환경에 이용하였다. Fig. 8은 블록을 이용하여 구성된 주행 환경을 무한대의 z 방향에서 바라본 모습이다. 그림에서 빗금친 정사각형이 가장 큰 단계의 블록이며 이 블록은 중간 단계의 블록 4개로 구성된다. 각각의 중간 단계의 블록은 25개의 최소 단위의 블록을 포함한다. 가장 큰 블록은 가로, 세로의 크기가 500 m의 정사각형이 되고, 한 변의 크기가 250 m인 두 번째 단계의 정사각형 블록 4개를 포함한다. 최소 단위의 블록은 한 변의 크기가 50 m인 정사각형이다. 각 단계의 블록은 왼쪽 아래에서부터 시작하여 00, 10, 20... 으로 번호가 매겨진다. 두 자리의 번호는 각각 x와 y의 좌표값을 나타낸다.

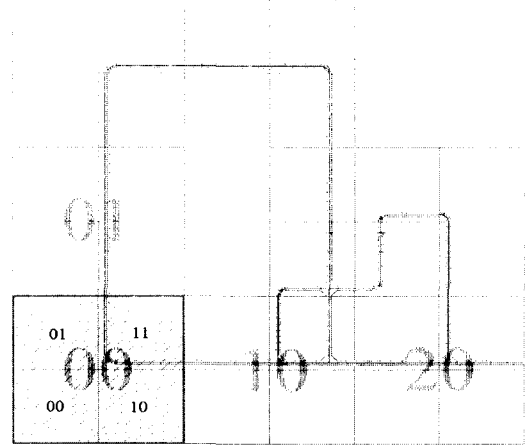


Fig. 8 Driving environment block

굴삭기 주행 환경은 Table 2와 같이 정해진 규칙에 따른 주행 환경 파일을 제작하여 읽어 들여 사용한다. 한 개의 큰 블록(500X500)은 여러 개의 중 블록과 작은 블록으로 구성되어 있으며, 블록의 포함 관계를 'end'문장을 사용하여 표현하였다. 실질적인 주변 환경의 요소, 즉 도로, 교차로 등은 최소 단위의 블록에서 결정된다. Table에서 보는 바와 같이, 번호가 (1 0)인 중간 크기의 블록은 커브와 3개의 도로 및 교차로로 구성되었다. 각 블록의 번호는 자신이 소속된 블록에 대한 상대적 값이므로 소속 블록의 번호만 바꿈으로써 전체 영역을 이동시킬 수 있으며, 주변 환경을 첨가하거나 확장할 때에도 간단하게 수정할 수 있다. 또한 타일 접근법은 전체의 환경을 검색하여야 하는 반면 블록 개념은 포함하고 있는 블록만 검색하기 때문에 고속 검색이 가능하다. 주행 환경 파일은 텍스트로 구성되어 작성과 수정이 용이하고, 별도의 컴파일 없이 사용할 수 있는 장점이 있다.

Table 2 Driving environments file

filename : road.map
500x500 1 1
250x250 1 0
50x50 0 0 curve 2
50x50 1 0 road 2
50x50 2 0 road 2
50x50 3 0 inter 5
50x50 4 0 road 2
:
250x250 1 1
50x50 0 0 road 2
50x50 1 0 road 2
50x50 2 0 road 2
50x50 3 0 curve 1
50x50 4 0 inter 5
50x50 5 0 curve 2
end
end500
endall

주변의 대상이 실제와 같이 많으면 많을수록 현실감이 있고 시뮬레이터 운영을 위해서는 실시간으로 주변 대상을 처리하여야 한다. 이러한 상반되는 요구를 절충하는 범위 내에서 주변의 대상을 표현한다. 우선 최소한의 주변 대상만을 표현하고, 시스템이 초당 15~30 프레임 정도의 화면을 처리할 수 있는 범위 내에서 대상의 개수를 늘려 나간다. 초기 그래픽 화면은 처리 속도를 고려하여 위해 주변의 모든 대상은 생략하고, 도로와 가로등만이 표현된다. 사용자의 선택에 따라 건물과 자연물을 추가하고 배경을 이용할 수 있도록 하였다.

이용하였다. 난수를 이용한 프랙탈 함수는 간단한 알고리즘으로 계속 다른 형상을 만들어내기 때문에 자연현상을 나타내기 위해 적절하다. 프랙탈 함수를 이용하면 대부분의 자연물을 나타낼 수 있지만, 계산량이 많아 속도가 느리므로 제한적으로 사용하였다. 도로 주위에 놓일 건물들은 간단한 다각형의 조합으로 만들었으며, 텍스처 맵핑 기법을 도입하여 표현함으로써 현실감을 더욱 높이도록 하였다.

자연 대상물을 표현하기 위하여 프랙탈 함수를

#### 4. 그래픽 시뮬레이터 통합 환경

그래픽 시뮬레이터를 운영하는 통합 환경은 모티프(Motif)를 이용하여 윈도우 기반으로 작성하였다. 윈도우는 사용자가 키보드를 사용하지 않더라도 마우스를 이용하여 아이콘이나 메뉴를 조작함으로써 간단하게 프로그램을 수행할 수 있다. 모티프는 X-윈도우 시스템에서 구동되는 다양한 기능이 함수 형식으로 구성된 객체지향적 프로그램 라이브러리며 하드웨어 환경에서도 호환성이 뛰어나며, ANSI C언어 및 X, Xt 라이브러리 등과 호환이 가능하며 그래픽은 모티프 윈도우에 드로잉 영역을 생성하여 OpenGL을 사용하여 나타낼 수 있다.

그래픽 시뮬레이터 통합 환경은 주행환경, 굴삭기, 인체 모델들을 그래픽 처리할 뿐만 아니라 굴삭기 주행 그래픽 시뮬레이션을 구현하는 기능을 가지고 있다. 또한 동역학 결과를 학습시킨 신경망을 포함하고 있다. Fig. 9에 개발된 환경의 사용자 인터페이스가 보여진다.

통합환경 운영 모듈을 생성하기 위해 셀을 많이 사용하지 않고 메모리를 생성하고 해제시키는 방법으로 하나의 메인 셀에서 여러 가지 모티프 위젯을 교환하는 방식을 구현하였다. 서브 셀에는 운전자의 시점에서 주행환경을 보여 주기 위해 그래픽 영역이 만들어지며, 메인 셀에는 사용자가 통합 환경을 운용하고 관리하기 위한 GUI가 생성되고 굴삭기, 인체 모델, 주행환경 지도 등의 그래픽 처리도 보여줄 수 있도록 하였다.

#### 5. 그래픽 시뮬레이션 수행

실시간 그래픽 처리에서 동역학 해석은 많은 계산량이 요구되므로 이를 빠르게 처리하기 위해서는 새로운 동역학 알고리즘의 개발과 모델링의 단순화를 통해 계산량을 줄이는 방법과 성능이 좋은 컴퓨터를 이용하는 방법이 있으나 새로운 알고리즘 개발과 모델링에 많은 시간이 소요되고 고성능 컴퓨터의 이용에도 많은 비용이 들게 된다. 본 연구에서는 동역학 해석을 실시간으로 처리하는 방법의 하나로 오프라인으로 동역학 알고리즘을 이용하여 계산된 결과를 신경망 중에서 백프로퍼게이션(back propagation) 학습 알고리즘을 통해 학습을 수행한 후 이 학습된 결과를 사용하여 임의의 입력에 대한 원하는 출력을 구하는 방법을 도입하였다.



Fig. 9 Graphic user interface

Table 3 Training input data vectors

Time (sec)	Steering angle (rad.)	Velocity (m/s)	Position in x-direction (m)	Velocity (m/s)	Position in x-direction (m)
4.0	0.0000	5	19.49	7	27.29
4.1	0.0031	5	19.97	7	27.96
4.2	0.0062	5	20.46	7	28.64
4.3	0.0093	5	20.94	7	29.32
4.4	0.0124	5	21.43	7	30.00
4.5	0.0155	5	21.92	7	30.69
4.6	0.0176	5	21.41	7	31.37
4.7	0.0201	5	22.59	7	32.05
4.8	0.0223	5	23.38	7	32.73
4.9	0.0243	5	23.87	7	33.41
5.0	0.0260	5	24.35	7	34.10
5.1	0.0274	5	24.84	7	34.78
5.2	0.0285	5	25.33	7	35.46
5.3	0.0293	5	25.81	7	36.14
5.4	0.0298	5	26.30	7	36.82
5.5	0.03	5	26.79	7	37.50

그래픽 굴삭기 시뮬레이터에 운전자의 조작에 의해 입력으로 들어가는 값은 핸들의 회전각과 가속페달, 브레이크 페달의 각도이다. 이 값들은 시뮬레이터에 설치된 엔코더에 의해 값이 측정되어 입력된다. 여기서 핸들의 회전각과 바퀴의 조향각

의 관계를 이용하여 조향각의 각도를 구할 수 있다. 굴삭기의 속도는 가속 페달과 브레이크 페달의 각도를 이용하여 계산이 가능하다. 굴삭기 시뮬레이터의 운동 시스템을 구동하고 그래픽 화면 재현을 위해 필요한 출력값은 굴삭기 중심의 위치 ( $x, y, z$ )와 브라인트 앵글로 표시되는 방위( $\theta_x, \theta_y, \theta_z$ )의 6개 값이다. 신경망 학습은 주어진 데이터를 이용하여 가중치와 노드의 바이어스 값을 구하는 방법으로 수행하였다. 학습률  $\eta$ 와 모멘텀 비  $\alpha$ 는 학습이 진행되는 동안 조절되도록 하였다.

굴삭기 주행에서 차선 변경, 코너링, 범프 통과의 3가지 운전 상황에 대해 신경망 접근법을 도입하였다. 각각의 운전 상황에 대해 학습을 시킨 후, 온라인 시뮬레이션 수행 중 운전 상황의 이벤트가 발생하면 그 운전 상황에 맞는 신경망의 학습 결과를 불러 사용한다. 본 논문에서는 굴삭기가 주행 중에 차선을 변경하는 경우에 대하여 신경망을 이용하여 학습한 결과를 논하겠다. 여기서 굴삭기는 평면을 주행하는 경우로 한정하였으며 굴삭기 모델은 굴삭기를 단순화하여 10개의 물체로 구성되어 있고 13의 자유도를 가지는 모델을 사용하였다. 굴삭기의 차선 변경은 굴삭기가 정지한 상태에서 일정한 속도로 앞으로 전진하다가 4초부터 10초까지 6초간 바퀴의 조향각이 사인그래프 형태로 변화하는 것으로 가정하였다. 진행속도는 3, 5, 7 m/s의 3가지 경우이고 조향각의 진폭은 0.03, 0.05, 0.07 rad의 세 가지 경우로 이 데이터를 조합하여 총 9가지 경우를 학습하였다. 학습에 이용될 입력 데이터는 위의 9가지 경우에 대해 동력학 해석을 수행한 결과이다. 학습을 위하여 사용된 데이터 중 속도가 5 m/s와 7 m/s의 두가지 경우에

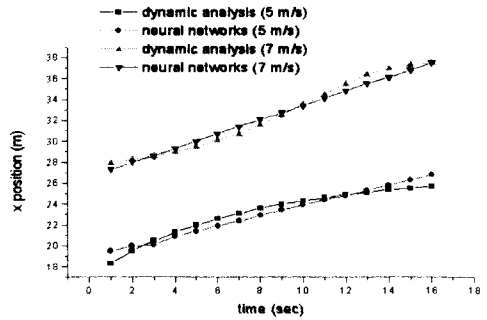


Fig. 10 Comparison between the dynamic analysis and neural network results



Fig. 11 Simulation of steering

동력학 해석 결과 중에서 전진방향(x 방향)의 위치 데이터의 일부가 Table 3에 보여진다. 신경망 접근법을 이용하여 학습한 결과의 가중치와 바이어스 값은 Table 4와 같으며, 초기에 설정한 학습 비는 0.5 이다. 동력학 해석을 수행한 결과, 즉 학습의 기준 값과 신경망을 이용하여 학습을 통해 구한 결과를 비교하면 Fig. 10과 같다. Fig. 10에서 보면 동력학 해석을 수행한 결과와 신경망 학습을 수행한 결과와 비슷함을 알 수 있다. 속도가 5 m/s와 7 m/s인 두 경우 모두 같은 경향을 보여 오차는 10% 안쪽으로 나타나며, 이 오차의 범위는 시뮬레이터의 구동과 그래픽 작업에 큰 영향을 미치지 않는다.

Fig. 11에 신경망의 동력학 데이터를 이용하여

Table 4 Weight matrix and biases  
(i: input, h: hidden, o: output)

Layer	Node	$W_{ji}$ (i: Preceding)							
	i j	1	2	3	4	5	6	7	8
i→h	1	1.75	0.40	0.75	0.75	2.05	0.75	0.75	2.05
	2	2.72	4.78	2.86	5.28	3.21	2.86	5.28	3.21
h→o	1	12.3	11.5	15.0	10.9	6.0	15.0	10.9	5.9
Biases	Hidden	-5.08	2.43	-5.30	3.35	4.95	-5.30	3.35	4.95
	Output	-2.78							



시뮬레이션을 구현하여 주행 과정을 중첩하여 나타낸 것이다. Table 5에는 굴삭기 차선 변경 시뮬레이션의 처리 속도를 여러 경우에 대하여 실험하여 비교한 것이 보여진다. 포함 모델의 수가 증가할 때마다 그래픽 처리 속도는 떨어지지만 모든 모델들 즉, 완전한 주변 환경과 인체 및 굴삭기 등이 포함된 경우에 초당 20 프레임으로서, 그래픽 처리가 현실감이 느껴지는 실시간 그래픽 시뮬레이터를 위해 필요한 초당 15 프레임의 화면 처리 범위에 있음이 보여진다. 그러나, 동역학 계산을 온라인으로 처리하여 시뮬레이션을 수행한 결과 초당 6 프레임 정도의 처리 속도가 보여졌다. 즉 신경망 접근법이 온라인으로 처리하였을 때와 비교하여 그래픽 처리가 3.3배정도 빠르게 처리됨을 알 수 있었다. 이런 그래픽 처리 결과는 온라인 동역학 계산 처리 시간을 줄였을 뿐만 아니라 굴삭기 주행중 실시간 주행 환경의 처리를 위해 상세 레벨 조정 및 인체 모델의 시각점에 의한 환경 처리 기법의 도입으로 가능하였다.

Table 5 Speed of graphic processing

Model	Process frame (frame)	Task time (sec)	Processing speed (frame/sec)
Road	3276	99.98	32.77
Rd. - Excavator	2674	99.99	26.74
Rd. · Exc. - Building - Mountain	2094	99.99	20.94
Rd. · Exc. · Bdg. · Mt. - Background	2046	99.99	20.46

## 6. 결론

본 연구에서는 굴삭기 시뮬레이터의 그래픽 실시간 처리가 가능하도록 굴삭기 모델, 인체 모델, 주변환경 모델을 개발하였다. 또한 굴삭기 주행 동역학 해석을 온라인으로 처리하지 않고 신경망 접근법을 이용하여 그래픽 처리 시간을 단축하였다. 굴삭기 모델은 주행 상황 위주로 제작되었지만, 작업 상황에서도 사용 가능하도록 계층적인 구조로 구성되었으며, 인체 모델을 이용하여 객관성 있는 주행 환경 구현을 가능하게 하였다. 블록개념의 주행환경을 텍스트 형태의 파일로 작성하는 방법을 제시하여 사용자가 간단하게 구성할 수 있도록 하였으며, 텍스처 매핑을 통한 현실감 재현과

그래픽 처리 속도 향상을 위해 모델링 기법을 개발하였다. 그래픽 통합 환경을 구성하여 환경내에서 개발된 그래픽 모델을 이용하여 굴삭기 주행 상황을 시뮬레이션을 구현하였으며, 시뮬레이션 결과 이들 모델과 신경망 접근법의 효용성을 보여 주었다.

현재 인체 모델의 이용은 시야 평가를 위한 시점 위치와 시선 방향을 입력받기 위한 수단으로만 사용되었다. 앞으로의 연구에서는 굴삭기의 작업 시뮬레이션에서 인체 모델을 사용하여 각 관절의 힘, 충돌시 상해 정도 등을 수행하고자 한다.

## 후기

본 연구는 과학재단 "특정기초 연구과제 (97-0200-10-01)"의 지원으로 이루어졌으며, 이에 관계자 여러분께 감사드립니다.

## 참고문헌

1. Stickney, M. E., "Computer Simulation: The Beginning of a New Era," SAE Paper 930885, 1993.
2. Bartollini, B., "The General Motors Driving Simulator," SAE Paper 940176, 1994.
3. Greenberg, J. A. and Park, T. J., "The Ford Driving Simulator," SAE Paper 940179, 1994.
4. Cremer, J., Kearney, J., and Papelis, Y., "Driving Simulation: Challengers for VR Technology," IEEE Computer Graphics and Applications, pp. 16~20, 1996.
5. Kuhl, J., Evans, D., Papelis, Y., Romano, R., and Watson, G. S., "The Iowa Driving Simulator: An Immersive Research Environment," IEEE Computer Magazine, pp. 35~41, 1995.
6. Papelis, Y. E., "Terrain Modeling on High-Fidelity Ground Vehicle Simulators," <http://www.ccad.uiowa.edu/research/ids>, The University of Iowa., 1997.
7. Drosdol, J. and Panik, F., "The Daimler-Benz Driving Simulator: A Tool for Vehicle Development," SAE paper 850334, 1985.
8. Käding, W. and Hoffmeyer, F., "The Advanced Daimler-Benz Driving Simulator," SAE paper 950175, 1995.

9. 이승호, 김성덕, "PC 기반형 자동차 운전 연습기 개발," 제어·자동화·시스템공학회지, pp. 415~421, 1997.
10. 조준희, 박경균, 이운성, 김정하, "차량 시뮬레이터의 시각 및 음향 시스템 개발," 한국자동차공학회지, pp. 476~481, 1997.
11. 손권, 이희태, "인체모델을 이용한 중장비 운전실 설계용 CAD 프로그램," 대한기계학회 논문집, pp. 3525~3537, 1997.
12. Cheng, H., Obergefell, L., and Rizer, A., Generator of Body Data (GEBOD) Manual, Wright-Patterson Air Force Base, Ohio, 1994.
13. Obergefell, L., Gardner, T., Kaleps, I., and Fleck, J., Articulated Total Body Model Enhancements, Wright-Patterson Air Force Base, Ohio, 1988.