

PC기반의 가상현실을 이용한 굴삭기 조작 시뮬레이터 개발

이세복*, 김인식*, 조창희* 김성수**

Development of a PC based Simulator for Excavator Manipulation using Virtual Reality

Se-Bok Lee*, In-Shik Kim*, Chang-Hee Cho*, Sung-Soo Kim**

Key Words: Virtual Reality, Excavator, Simulator

Abstract

A low cost PC based simulator for excavator manipulation has been developed using virtual reality technology. The simulator consists of two joystick input devices, server and client PCs, an excavator kinematics module, and a graphic rendering program Open Inventor. In order to use two joysticks in the PC window environment, multi-thread programing with network protocol TCP/IP has been used. To provide realistic view to the operator, CAD program Pro/Engineer and 3D modeller have been employed to create 3D part geometry of the manipulator and virtual environmental geometries. Those geometries also have been transformed and imported to the Open Inventor. The Simulator developed is to be improved for more realistic excavator operational training.

1. 서 론

가상 현실(Virtual Reality)이란 다양한 컴퓨터 인터페이스를 이용하여 사용자의 몸짓과 음성 등을 감지하여 사용자의 의도를 추출하고 이에 따라 적절한 응답을 출력함으로써 사용자로 하여금 컴퓨터가 생성하는 가상 환경을 직접 체험하게 하는 제반 기술을 가리킨다. 가상 현실 기술은 90년대의 주요 기술로서 컴퓨터의 고속, 대용량화, 지능화에 더불어 3차원화를 이루도록 하는 기술이며 기타 산업에의 파급효과가 매우 큰 기술이다. 시뮬레이션 기술은 컴퓨터의 저가·고기능화로 인해 제품의 설계에서부터 게임에 이르기

까지 폭넓게 사용되고 있다.

이러한 가상현실 기법은 시뮬레이터에 적용되어, 항공기 조정훈련 시뮬레이터, 감성공학용 자동차 시뮬레이터 등에 사용되어지고 있다.⁽¹⁾⁽²⁾

이러한 시뮬레이터는 주로 고가의 그래픽 전용 컴퓨터와 실시간 계산 컴퓨터, 운동 재현기, 전체 시스템을 제어하는 컴퓨터 등 고가의 장비로 구성되어있다. 이러한 고가의 가상현실 부과장치와 운동재현을 겸비한 시뮬레이터와는 달리, 가상현실 기술은 건설중장비 조작훈련기에도 적용할 수 있다. 건설 중장비조작은 고정된 기반에서 주로 진행되므로 운동재현을 고려하지 않은 고정기반 시뮬레이터가 적합하다. 가상현실 기술을 실질적인 건설장비 조작훈련에 적용하려면 보다 경제적인 저가형의 시뮬레이터 개발이 필요하다.

본 논문에서는 건설 중장비인 굴삭기의 조작훈련을 위하여 저가형 PC 기반의 굴삭기 조작부 시뮬레이터 구현 연구에 목적이 있다. 실제 굴삭

* 충남대학교 대학원
** 회원, 충남대학교 메카트로닉스공학과
E-mail : sookim@hanbat.chungnam.ac.kr
TEL : 042)821-7782 FAX : 042)823-4919

기의 조정과 같이 두 개의 조이스틱을 사용되었으며, CAD 프로그램인 Pro/Engineer⁽³⁾와 3D MAX⁽⁴⁾ 프로그램을 사용하여 굴삭기 형상과 가상환경을 제작하였다. PC상에서 그래픽 렌더링 도구인 OpenInventor를 기반으로 시뮬레이터를 구현하였다.

2. 시스템 구성

시뮬레이터의 하드웨어 구성은 Fig. 1과 같이 기구학적 프로그램과 그래픽 프로그램, 조이스틱 인터페이스를 담당하는 호스트 컴퓨터, 그리고 두 개의 조이스틱 사용으로 인한 다른 하나의 조이스틱 입력을 받기 위해 사용되는 클라이언트 컴퓨터로 구성된다. 여기서 호스트 컴퓨터는 Pentium II 350MHz CPU와 8M의 그래픽 카드, RAM 128M 사양의 시스템이다.

시뮬레이터는 굴삭기 조작부 동작법에 따라 2개의 조이스틱을 조정하면, 클라이언트 컴퓨터를 통하여 조이스틱 인터페이스로 보내진 신호에 의해 굴삭기의 회전각 및 유압 실린더 길이가 계산된다. 이러한 길이 변화와 회전각의 변화에 따라 굴삭기를 구성하고 있는 각 암, 붐, 버킷 링크들의 위치와 자세가 호스트 컴퓨터에서 기구학적으로 계산된다. 또한 계산된 위치와 자세는 3차원 형상에 인가되어, 호스트 컴퓨터에 있는 그래픽

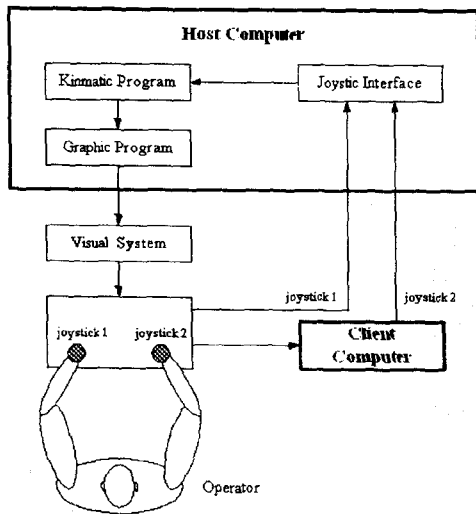


Fig. 1 시스템 구성

렌더링 프로그램에 의해서 가시화 되어 모니터를 통해 조작자에게 보여진다. 조작자는 굴삭기의 움직임에 따라서 계속적으로 조이스틱을 조정하게 된다.

3. 굴삭기 CAD 모델링(Pro/E)

가상 현실환경에서 보다 사실적인 굴삭기 형상을 구현하기 위해 CAD 프로그램인 Pro/E⁽³⁾로 굴삭기를 모델링 하였다. Fig. 2는 총 17개로 구성된 굴삭기의 파트 중 붐 및 암의 파트 모델을 나타내고 있으며 Fig. 3에서는 Pro/E에서 조합된 굴삭기 모델을 나타내고 있다. 이 Pro/E 모델은 VRML 포맷으로 변환된 후 가상 환경으로 불러들여 사용된다.

또한 이 Pro/E 모델은 범용 기구 동역학 해석 프로그램인 DADS⁽⁵⁾와 그 인터페이스 프로그램인 DADS/Pro 프로그램을 통하여 기구학 해석에 이용되었으며, 개발된 기구학 프로그램의 검증에 사용된다.

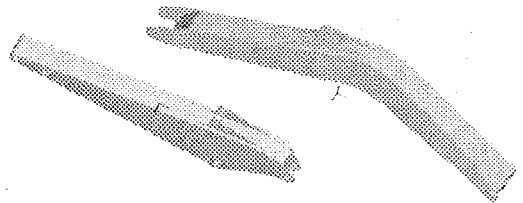


Fig. 2 암, 붐의 파트 모델

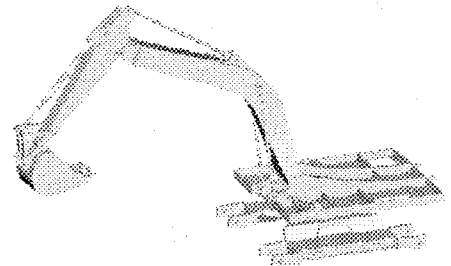


Fig. 3. Pro/E에서 모델링한 굴삭기

4. 굴삭기 기구학

4.1 기구학 모듈 생성

굴삭기의 기구적 해석에서는 유압 실린더의 길이 변화에 따른 각 물체의 위치와 자세를 계산한다. Fig. 4는 유압 실린더로 연결된 붐과 암의 도식적인 그림을 나타내고 있다. 유압실린더의 길이 변화에 따른 붐과 암 사이의 상대적인 회전 각 변화는 다음과 같이 Cosine 제 2법칙을 사용하여 구할 수 있다.

$$\theta = \arccos\left(\frac{A^2 + B^2 - C^2}{2AB}\right) \quad (1)$$

각 부분 위치는 구해진 사잇각 θ 의 함수로 표시되는 4×4 변환 행렬⁽⁶⁾을 이용하여 아래와 같은 식으로, 순차적으로 베이스 물체로부터, 붐, 암, 버킷으로 구할 수 있다.

$$\begin{bmatrix} P_X^A \\ P_Y^A \\ P_Z^A \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_X^B \\ P_Y^B \\ P_Z^B \\ 1 \end{bmatrix} \quad (3)$$

이러한 순환적인 공식에 의해 C 언어를 이용하여 기구학 프로그램이 생성되었다.

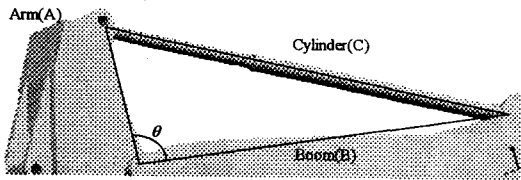


Fig. 4 붐, 암, 실린더의 연결도

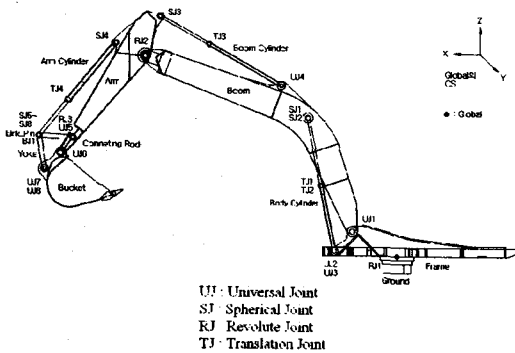


Fig. 5 굴삭기 DADS 모델

4.2 기구학 모듈 검증

기구학 모듈은 기구 동역학 해석 프로그램인 DADS를 이용하여 검증하였다. Fig. 5는 굴삭기의 DADS 모델을 나타내고 있다. 굴삭기의 전체 시스템 구성은 Frame, Boom, Arm, Bucket, Connecting Rod, Yoke, Boom Cylinder, Arm Cylinder, Bucket Cylinder, Link_pin의 파트와 이들을 연결하는 Revolute Joint 3개, Spherical Joint 8개, Translational Joint 4개, Universal Joint 8개, Bracket Joint 1개로 모두 24개의 조인트로 구성되었다.⁽⁷⁾ 따라서 시스템은 4자유도를 가지며, 이에 따라 기구학 해석에서 4개의 Driver를 사용하였다.

Fig. 6에서는 버킷 위치 궤적을 나타내고 있으며 생성한 기구해석 모듈의 결과가 DADS의 해석 결과와 잘 일치함을 알 수 있다.

5. Man-Machine Interface

굴삭기는 2개의 조이스틱에 의한 조정에 의해 회전운동 및 조작부(Manipulator)의 운동이 가능하다.

본 연구에서는 대우중공업(HDOS 조작법)⁽⁸⁾에서 제작하는 굴삭기의 조작법을 기반으로 조이스틱의 조작을 구현하였다. 사용된 굴삭기 조작에 따른 붐, 암, 버킷, 선회의 4가지 동작법을 살펴보면 다음과 같다.

① 붐 : 오른쪽 조이스틱을 상·하로 움직임에 따라서 붐의 상승과 하강을 조정.

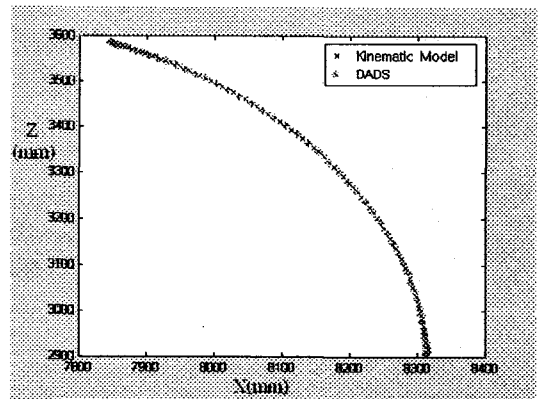


Fig. 6 버킷의 위치 궤적 비교

② 암 : 왼쪽 조이스틱을 상·하로 움직임에 따라서 암의 오프립과 덤프를 조정.

③ 버킷 : 오른쪽 조이스틱을 좌·우로 움직임에 따라서 버킷의 오프립과 덤프를 조정.

④ 선회 : 왼쪽 조이스틱을 좌·우로 움직임에 따라서 좌회전과 우회전을 조절.

이와 같은 동작을 하는 man-machine 인터페이스를 구현하기 위해서 Fig. 7과 같이 두 대의 조이스틱을 통신을 이용하여 두 대의 PC로 연결하였다.

조이스틱 입력 시 컴퓨터의 내부적인 동작은 윈도우에서 제공하는 DirectX⁽⁹⁾를 사용하여 외부 입력장치의 상태를 확인하고, TCP/IP의 통신 프로토콜을 사용하여 사용자로부터의 Feedback을 받게 하였다.

두 개의 입력은 서버와 클라이언트 관계를 이용하여, Visual C++⁽¹⁰⁾ 프로그램 상에서 계속하여 조이스틱의 정보를 얻고, 이를 서버에 보내기 위해서는 멀티스레드 방법을 사용하였다. 이는 윈도우 상에서 여러 가지 프로그램이 시분할을 이용하여 동시에 실행되는 것처럼 보이는 방식으로 하나의 프로그램 상에서 다른 프로그램을 실행시켜 새로 실행된 프로그램은 본 프로그램과 관계없이, 계속하여 자기 일을 수행 할 수 있게 하려는 목적이다. 따라서, 이를 사용하여 본 프로그램은 계속하여 조이스틱 정보를 사용하고, 서버 프로그램은 이 정보를 계속하여 서버 프로그램에 보내는 방식을 채택하였다.

프로그램의 구조는 서버 프로그램은 항상 실행되고 있다는 가정 하에 클라이언트 프로그램이 서버에 접속하여 하나의 커넥션을 맺고, 그후에 서버 프로그램은 계속하여 하나의 조이스틱을 폴링 하여 데이터를 얻으며, 클라이언트 프로그램은 폴링을 사용하여 데이터를 얻은 후 이를 서버

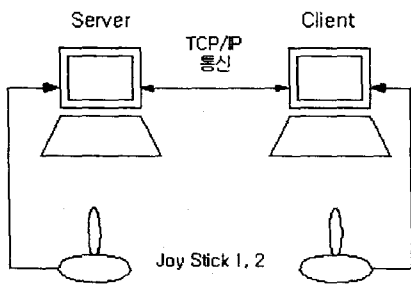


Fig. 7 조이스틱의 연결

프로그램으로 보내는 구조를 갖고 있다.

6. 가상환경 구축 및 시뮬레이터 구현

가상환경 속에서 굴삭기 작업을 할 수 있도록 하기 위해서는 가시화 렌더링 시스템을 이용하여 가상환경의 기본 화면을 형성한 후 조작자의 시야를 카메라 함수를 사용하여 결정해야한다. 시야 안에 보이는 가상환경은 3D MAX 가상환경 모델러를 사용하여 생성하였으며, CAD 프로그램인 Pro/E에서 제작된 굴삭기 조작부의 정교한 형상 모델을 가상환경에 VRML 파일로 변환시켜 도입하였다.

6.1 가상환경의 기본 화면 및 시야 설정

가상환경의 기본화면 설정을 위해서 본 연구에서는 3차원 렌더링 프로그램인 Open-Inventor⁽¹¹⁾⁽¹²⁾를 사용하였다.

가상 환경은 IVF Wizard를 사용하여 기본적인 화면을 형성하였다. 이 때 초기 화면의 설정은 다음과 같이 설정하였다.

윈도우의 크기-> 1024 × 768

화면 구성 -> Single Document

Viewer -> SoXtExaminerViewer

(SoXtExaminerView : Viewer component which uses a virtual trackball to view the data)

본 논문에는 SoPerspectiveCamera Node(Fig. 8)를 사용하여 가상환경 내에서 원근감을 나타낼

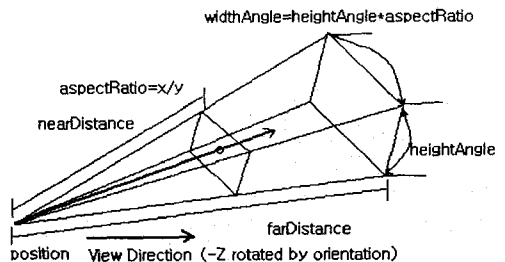


Fig. 8 View Volume and Viewing Projection for an SoPerspective Camera Node

수 있도록 하였으며, 사용자 시야에서 본 화면을 얻기 위해서 다음과 같은 Camera Class의 멤버 함수를 사용하였다.

```
myCamera -> position.setValue(-1500, 4000, 2000);
```

다음으로 회전하는 시야를 얻기 위해서 아래와 같은 Camera Node에 이름을 부여하여 Camera의 설정을 다른 함수에서 변경할 수 있도록 하였다.

```
myCamera -> setName("mycamera_rotation");
```

6.2 가상환경 구현 및 CAD 형상 도입

가시된 렌더링 시스템으로는 OpenInventor를 사용하여 GUI를 구성하고 가상환경과 굴삭기 형상을 VRML 형식으로 도입하여 기구적인 해석 결과를 통해 각 굴삭기 조작부의 각 부분의 위치, 자세를 실시간으로 가시화 한다.

가상환경은 3D MAX environment generator와 산 모양을 생성해 주는 플러그 인을 사용하여 제작하였다. Fig. 9는 산을 포함한 가상환경의 모습이다.

Pro/E에서 제작된 굴삭기 조작부 형상을 OpenInventor 환경에 도입하기 위해서 먼저 굴삭기 조작부를 움직이는 최소단위로 분리하여 각각을 하나의 VRML로 변환하였다. Pro/E에서 모델링 된 각 Body는 정의된 좌표계의 위치나 방향이 Inventor에서의 좌표계가 일치하지 않는다. 따라서 Pro/E에서 생성된 파일은 다시 3D MAX에서 읽어들이 좌표계의 위치와 방향을 쉽게 변경시켜야 한다.

이렇게 좌표계를 수정하여 도입된 형상들을 각

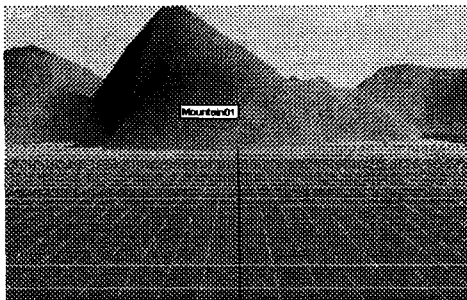


Fig. 9. Environment View

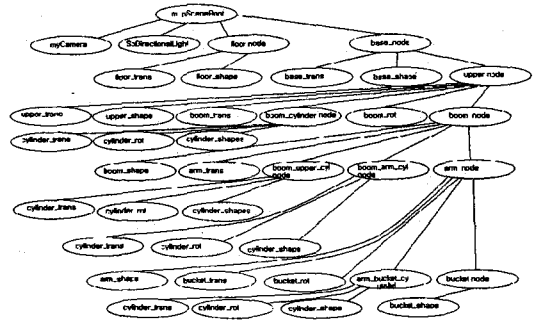


Fig. 10 객체 사이의 관계

개체 사이의 계층구조(Hierarchy)를 고려하여 node를 설정하여 전체 가시화 시스템을 완성하였다. Fig. 10은 굴삭기 전체 모형을 Inventor에서 나타내기 위해서 만든 계층구조이다.

3D MAX에서는 두 가지 version의 VRML 포맷을 지원하는데, Inventor에서는 낮은 version의 VRML 1.0 만을 읽어들이기 때문에 환경에 도입 시에 오브젝트의 해상도가 떨어지는 문제가 있었다.

6.3 시뮬레이터 구현

Fig. 11은 전체 시스템의 데이터 흐름도를 나타내고 있다. 6.1절에서 설명한 방법에 의해 조이스틱 입력을 받아들이고 신호를 실린더의 길이로 변환하여 기구해석을 한 후, 각 물체의 위치를 결정한다. 그리고 이 정보가 OpenInventor의 가시화 시스템으로 전달되어 화면을 통해 조작자에게 시각적으로 전달된다. Fig. 12는 구현된 PC 기반의 굴삭기 조작 모습을 보여주고 있다.

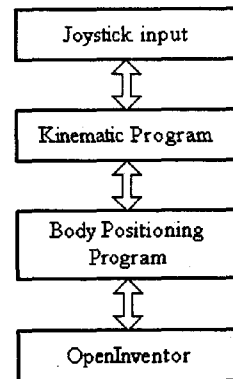


Fig. 11 Data Flow

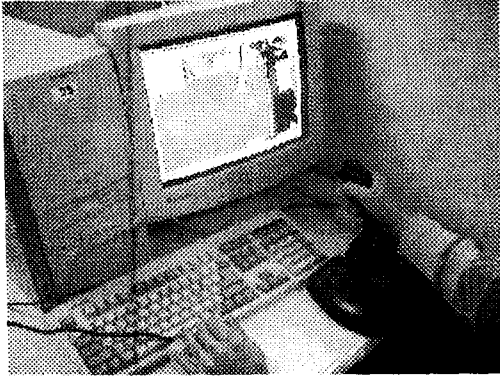


Fig. 12 시뮬레이터 조작 모습

5. 결론 및 추후 연구

본 연구를 통해 CAD 프로그램인 Pro/E를 이용하여 정교한 굴삭기 형상모델이 생성되었고, 3D MAX를 이용하여 가상환경을 생성하였다. 이러한 형상모델을 Inventor 환경으로 도입된 후 계층화하여 가시화 되었다. 또한, TCP/IP 네트워크를 통한 인터페이스로 두 개의 조이스틱의 man-machine interface를 구현하였다. 시스템 통합하여 저가형 PC 기반의 굴삭기 조작부의 시뮬레이터를 구현하였다.

현 시스템의 문제점으로 첫째, 기구해석 만으로는 굴삭기 움직임이 실제감이 부족하였고, 둘째 OpenInventor가 VRML 형식을 받아들일 때 이미지의 정도가 낮아지는 문제와 셋째, OpenInventor의 일반적인 8Mb 그래픽 카드사양의 PC에서의 실시간 그래픽 처리능력의 제한성 등을 지적할 수 있다.

위에서 언급한 문제점을 해결하기 위해서 앞으로 수행되어야 할 추후 연구는 실시간 굴삭기의 조작의 현실감을 높이기 위해 실시간 동역학 시뮬레이션 프로그램 개발과 collision detection을 통한 버킷의 지면과의 상호작용을 느낄 수 있도록 하는 힘 반향 조이스틱 구현이다.

또한 OpenInventor 환경을 Texture Mapping 과 Graphic 처리가 뛰어난 OpenGVS⁽¹⁴⁾ 환경으로 가상환경 재구성도 필요하다.

6. 참고 문헌

- (1) 이호근, 이은용, 허기봉, 천찬원, 고준수, 1996, "비행성 평가 시뮬레이터 개발", 한국항공우주학회지, 24, 5 pp. 157~164.
- (2) J.S. Freeman, et al., 1995, "Iowa Driving Simulator : An Implementation and Application Overview", SAE paper 950174.
- (3) Pro/E manual; Parametric Technology Corporation.
- (4) 이명수, 김재규, 김국선, 1999, "3D MAX R2", 영진출판사.
- (5) DADS manual, 1997, CADSI
- (6) John J. Craig, 1989, "Robotics", Addison-Wesley.
- (7) 유완석 외 6명, 1991, "굴삭기의 도로 시뮬레이션 및 연결부의 반력계산", 1차년도 최종보고서, 부산대학교 기계기술연구소.
- (8) 안정기, "굴삭기 운전 기능사", 세계 출판사
- (9) Barga Donnelly, "Inside Direct X", 영진 출판사.
- (10) 이상엽, "Visual C++ programming Bible", 영진 출판사.
- (11) Josie Wernecke, 1994, "The Inventor Mentor", Addison-Wesley.
- (12) Joie Wernecke, 1994, "Open Inventor Architecture Group", Addison-Wesley.
- (13) Richard S. Wright Jr, Michael Sweet, "OpenGL Superbible", 도서출판 에프원.
- (14) OpenGVS V4.2 manual, 1998, Quantum3D, Inc.