웹 기반 로봇 시뮬레이터

A Web-Based Robot Simulator

홍 순 혁, 이 상 현, 전 재 욱, 윤 지 섭
(Soon-Hyuk Hong, Sang-Hyun Lee, Jae Wook Jeon, and Ji Sup Yoon)

Abstract : According to the advancement of web related technologies, many works on robots using these technologies, called web-based robots, have been attempted. Web-based robots enables sharing of expensive equipments as well as control of remote robots. But none of the existing methods about web-based robots include robot simulators in their web browser, which transfer appropriate information of a remote place to the local users. In this paper, a web-based robot simulator is proposed and developed to control a remote robot by using the web. The proposed simulator can transfer the 3D information about the remote robot to the local users by using 3D graphics, which has not been previously developed. Also, it sends the camera image of a remote place to the local users so that the users can use this camera image as well as 3D information in order to control the remote robot.

Keywords : web-based robots, robot simulator, 3D graphics, camera image

I. 서론

웹 기술이 발전함에 따라서 현재는 어느 곳에서나 세계 도처의 정보를 이용할 수 있게 되었다. 로봇 분야에 있어서도 웹 기술은 새로운 분야를 탄생시켰는데 바로 웹 로봇 혹은 인터넷 로봇이란 분야이다.

웹 로봇은 원격지의 로봇을 웹 상에서 제어하고자 하는데 목적을 두고 있으며, 또 다른 중요한 목적은 로봇과 같이 값비싼 장비를 여러 사람들이 연구용이나 교육용으로 공유하고자 하는데 있다. 즉 값비싼 장비를 효율적으로 사용하고자 하는데 목적이 있는 것이다. 최근 이러한 웹을 이용한 원격제어가 많이 시도되고 있는 실정이며, 서비스 로봇과 교육용 로봇 그리고 앞으로 주목받게될 오락용 로봇분야에서도 활발히 연구가 진행되고 있다.

웹 로봇은 나사(NASA)의 화성 프로젝트의 일환으로 처음 소개되었고 아직까지 발전 초기 단계에 있기 때문에 여러 가지 면에서 부족한 점이 많다. Goldberg [1]는 웹 브라우저 내에서 카메라 이미지를 보여주는 부분과 사용자의 로봇제어 부분을 구성하여 로봇을 제어하는 웹 로봇 시스템을 제안하였다. 여기서는 사용자가 로봇의 운동 반경 내에 있는 임의의 위치를 마우스로 클릭을 하게 되면, 2차원 가상 로봇이 이동하도록 하였는데, 카메라 이미지는 로봇의 이동이 끝난 후에 보여지게 된다. 따라서, 사용자는 로봇의 이동경로를 예측하여 로봇을 제어할 수가 없는 단점이 있었다. Saucy와 Mondada [2]는 인터넷 환경에서 Khepera라는 이동로봇을 제어하기 위해 시스템을 개발하였는데, 이 시스템은 웹 서버와 로봇 제어 서버, 비전 서버가 한 컴퓨터 내에 존재하기 때문에 웹 서버에 접속자수가 많으면, 다른 서버들은 제 기능을 다하지 못한다. 따라서, 이러한 문제점을 해결하기 위해서 웹 서버와 로봇 제어 서버의 분리는 필수적이라고 할 수 있다. 또한, 사용자는 웹 인터페이스를 통하여 로봇을 제어하게 되는데, 인터페이스가 너무 단순할 경우, 로봇을 제어하기 위한 충분한 정보를 제공하지 못한다는 단점과 너무 복잡할 경우에 실제 제어를 하는 경우에 조작이 어려울 수 있다는 단점이 있다. 따라서, 웹 상에서 로봇을 조작하기 위한 인터페이스는 사용의 용이성과 실제 사용상의 기능성을 고려하여 설계되어져야 한다.

Schulz [3]등은 박물관 내부를 안내해주는 가이드 로봇을 웹 상에서 제어하는 시스템을 개발하였다. 박물관을 직접 방문한 사람은 로봇에 장착된 컨트롤 패널을 이용하여 목적지를 입력하게 되면 로봇은 그 장소까지 안내를 하게 된다. 박물관을 직접 방문하지 못하는 사용자들은 웹 서버에 접속하여 로봇을 제어할 수가 있는데, 단순히 로봇에 탑재된 카메라의 정보만을 이용하여 로봇의 상태에 대해서는 알 수가 없다. Hirukawa [4]등은 원격지에서 로봇을 제어하기 위한 표준 인터페이스로 VRML(Virtual Reality Modeling Language)를 사용하는 시스템을 제작하였는데, 사용자가 마우스를 사용하여 가상의 공간을 네비게이션하는 기능을 제공하였다. 여기서는 VRML을 사용하기 위해서 전용 웹 브라우저가 필요하게 되며, 실제 시뮬레이션 기능 등은 제공하지 못하였다. 윤병준 [5]등은 웹을 이용하는 로봇 매니퓰레이터의 원격제어에 관한 연구에서 가상 로봇의 끝단에 대한 각도 변화를 시뮬레이션하였다. 웹 상에서 사용자 인터페이스를 구현하였고,

접수일자 : 2000. 3. 7., 수정완료 : 2000. 12. 15.
홍순혁, 이상현, 전재욱 : 성균관대학교 (hsh@micro.skku.ac.kr/risionsys@hanmail.net/jwjen@yurim.skku.ac.kr)
윤지섭 : 한국원자력연구소 극한환경 원격조작 기술개발팀(jsyoon@kaeri.re.kr)
※ 본 연구는 과학기술부의 원자력 연구 개발사업의 일환으로 수행되었습니다.

인터페이스의 조작에 의하여 가상의 로봇을 제어하고자 하였다. 그러나 이는 단지 가상 로봇을 제어하는 것이므로, 원격지의 실제 로봇을 제어하고자 하는 웹 로봇의 개념과는 차이점이 존재한다.

기존연구에서 살펴본 것처럼 웹 기반 로봇 시뮬레이터는 전무하다. 웹 로봇은 웹 상에서 로봇을 제어하는데 목적이 있으므로 실제 로봇을 제어하기 전에 사용자는 충분히 로봇의 동작특성을 분석해야 하는데 이는 시뮬레이터를 사용함으로써 가능하게 된다. 그러나 기존의 웹 인터페이스는 위에서 언급한 바와 같이 단순히 2차원 로봇을 보여주므로 사용자는 로봇에 대한 직관적인 이해가 떨어질 뿐만 아니라, 동작특성을 파악하기도 어렵다. 대부분의 웹 로봇 시스템은 이런 방식을 취하고 있으므로 웹 기반의 시뮬레이터는 절실한 상황이라고 할 수 있다.

웹 로봇은 기본적으로 웹 상에서의 로봇 시뮬레이터를 포함해야 하고, 웹 상에서 로봇을 제어해야 하는 것이기 때문에 사용자 인터페이스(User Interface)는 웹 브라우저 내에서 실행되어져야 한다. 웹 기반의 시뮬레이터를 사용하는 것은 가상의 로봇을 제어함으로써 실제 로봇을 제어하기 전에 제어하고자 하는 대상 로봇 시스템의 특성에 대하여 파악하고, 대상 시스템을 실제로 제어할 경우 효율적으로 제어할 수 있기 때문이다. 그리고 웹 로봇은 실제 로봇을 사용자가 웹 상에서 제어하고 제어 결과를 웹 상에서 확인할 수 있어야 한다. 제어할 대상 시스템은 원격지에 존재하기 때문에 사용자의 명령이 올바르게 전달되었는지 확인하기 위해서는 제어 결과를 사용자에게 보내주어야 한다. 결과를 확인하기 위해서 많이 사용하는 방법은 사용자의 명령을 실행한 후 로봇의 위치를 알기 위해 로봇의 끝단에 달린 카메라 이미지를 전송하거나 로봇의 전체적인 모습을 카메라 이미지로 전송하는 방법이 있다.

그러나 기존의 웹 로봇 연구에서 웹 기반 로봇 시뮬레이터를 개발하고자 하는 시도가 거의 없었으며, 이로 인해 사용자는 단순히 로봇의 특성을 예측하며 로봇을 제어해야 한다는 단점이 있었다. 따라서 로봇의 동작 특성을 이해하지 못해서 생기는 예측불허의 상황에 대하여 대비할 방법이 없게 된다는 것이다. 그리고 현재까지의 시뮬레이션 환경은 한 시스템에서 독립적으로 실행되어지는 시뮬레이터들이 대부분을 차지하였는데, 사용자는 고정된 위치에서만 시뮬레이터를 사용할 수 있기 때문에 사용자의 위치에 종속될 수밖에 없다.

본 논문에서는 기존 시뮬레이터의 한계점을 극복하고 웹 로봇의 개념에 따라 웹 기반의 로봇 시뮬레이터를 개발함으로써 사용자가 어느 장소에서나 시뮬레이터를 사용할 수 있도록 하였다. 그리고 웹을 이용하여 로봇을 제어하고자 하는 사용자가 실제 로봇을 제어하기 전에 로봇의 동작 특성을 충분히 숙지할 수 있게 하였다. 사용자는 웹 기반의 시뮬레이터를 통해서 제어 로봇의 동작 특성을 충분히 알 수 있을 뿐만 아니라 효율적으로 실제 로봇을 제어할 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2 장에서는 본 논문에서 제안한 전체적인 시스템 구성과 소프트웨어적인 구성, 웹 기반 로봇 시뮬레이터의 외형 및 기능, 로봇 컨트롤 서버의 외형 및 기능, 본 논문에서 사용한 통신 프로토콜에 관하여 설명하고, 3 장에서는 제안한 시뮬레이터를 이용하여 시뮬레이션을 수행하고 실제 시스템과의 연계 결과를 설명한다. 4 장에서는 본 논문에 관한 결론을 내리기로 한다.

II. 웹 기반 로봇 시스템

1. 전체 시스템 구성

본 논문에서 개발한 웹 기반 로봇 시뮬레이터(Web-Based Robot Simulator : WBRS)는 웹을 기반으로 하므로 웹 브라우저 내에서 실행될 수 있도록 프로그래밍 하였다. WBRS는 웹에 연결되어 분산 시스템과 다중 사용자를 고려하여 설계되었고, 전체적인 시스템 구성은 클라이언트/서버 구조로 되어 있다 [6]. 클라이언트는 웹 서버에 접속하여 동적 링크 라이브러리(Dynamic Link Library : DLL)형식으로 되어진 특별한 프로그램을 다운 받는다. 이 동적 링크 라이브러리는 클라이언트 시스템의 레지스트리에 등록되어져 사용자가 필요로 할 경우 동적으로 호출할 수 있다. 웹 서버는 사용자의 로봇 제어 명령을 전달하기 위해 로봇 제어 서버와 통신을 하게 된다. 사용자의 명령을 받은 로봇 제어 서버는 최종적으로 로봇에게 이를 전달한다. 전체적인 시스템 구성은 그림 1과 같다.

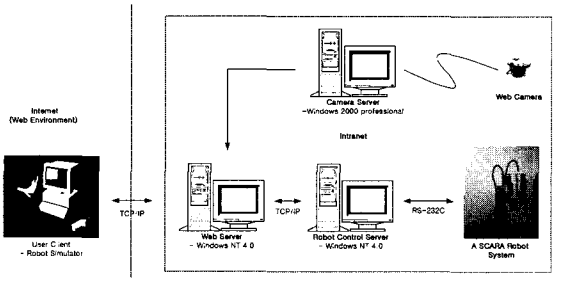


그림 1. 시스템 구성도.

Fig. 1. System architecture.

그림 1에서 보는 것처럼 사용자가 실제 로봇을 제어하기 위하여 시뮬레이터를 이용하여 로봇 제어 명령을 내리면 이는 로봇 제어 서버를 통해 실제 로봇에게 전달된다. 카메라 서버로 들어온 이미지는 웹 서버로 지속적으로 업로드되어 사용자들이 실제 로봇의 제어 결과를 웹 상에서 확인할 수 있게 하였다. 본 논문에서는 삼성 SCARA 로봇 FARA SM3를 시뮬레이션 할 대상 시스템으로 선정하였고, 직접 제작한 하드웨어를 사용하지 않고 삼성의 로봇 컨트롤러를 사용하여 로봇을 제어하였다. Windows 2000과 펜티엄 III 컴퓨터를 기반으로 WBRS의 개발환경을 설정하였고, 웹 브라우

저 내에서 실행되는 프로그램을 위해 Microsoft사의 ATL이라는 라이브러리와 산업계의 표준 그래픽 라이브러리로 자리잡은 OpenGL을 사용하였다.

2. 소프트웨어 구조

2.1 로봇 시뮬레이터 클라이언트

전형적으로 TCP/IP를 사용한 클라이언트/서버 구조는 한 시스템에서 독립적으로 실행되어지는 어플리케이션을 이용하는 방법과 웹 브라우저 내에서 어플리케이션이 실행되어지는 두 가지 방법이 존재한다. 전자의 방법은 사전에 프로그램을 배포해야 하는 문제점이 발생하게 되나, 후자의 방법은 웹 브라우저 내에서 실행되어지는 어플리케이션은 특정 웹 서버에 접속하면 해당 동적 링크 라이브러리가 시스템의 레지스트리에 등록되어지거나 애플릿이 실행되어지므로 프로그램을 사전에 배포할 필요가 없다. 본 논문에서는 후자의 방법을 채택하여 로봇 시뮬레이터를 개발하였다.

본 논문에서 개발한 웹 브라우저 내에서 실행되어지는 로봇 시뮬레이터는 내부적으로 크게 다섯 개의 모듈로 구성되어져 있다. 사용자의 입력을 받아들이는 GUI모듈, GUI모듈에서 받아들인 사용자의 마우스나 키보드 이벤트를 처리하는 모듈, 마우스 이벤트에 의해서 기하학적 변환을 가능하게 하는 모듈, 시뮬레이션을 가능하게 하는 시뮬레이션 엔진 모듈, 웹 서버와 통신하기 위한 TCP/IP 모듈 등으로 구성되어졌다. 기하학적 변환을 가능하게 하는 부분은 사용자가 가상 카메라의 뷰포인트를 제어하는 것으로 카메라 회전 버튼을 누르면 카메라가 회전하는 것처럼 뷰포인트를 변경시켜주는 것을 의미하며, 이동 버튼은 카메라가 좌우로 이동, 줌 버튼은 오브젝트가 고정되어 있고 카메라를 줌 인/아웃 하는 것과 동일하게 뷰포인트를 변경

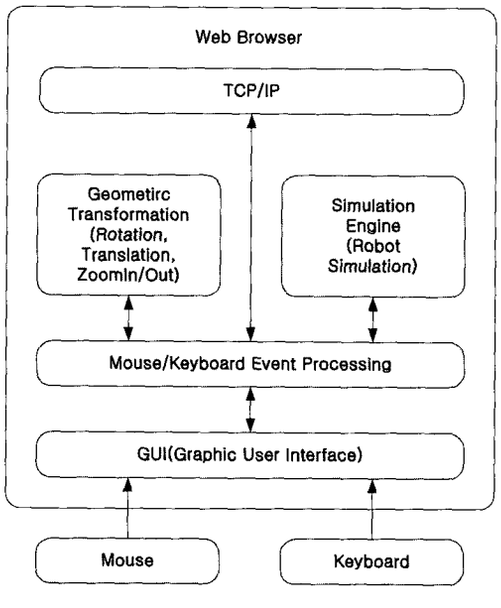


그림 2. 클라이언트의 소프트웨어 구조.
Fig. 2. Software architecture of client.

시키는 것을 의미한다. 클라이언트의 시스템 소프트웨어 구조는 그림 2와 같다. 그림 2에서 보는 것처럼 사용자의 마우스나 키보드 이벤트를 받아들이는 GUI 부분이 있고, 이벤트에 따라서 가상 카메라를 사용하여 뷰포인트를 변경하는 기하학적 변환부분과 시뮬레이션을 가능하게 하는 시뮬레이션 엔진부분의 상호 연동에 의하여 시뮬레이터가 동작을 하게 된다.

2.2 로봇 컨트롤 서버

원격지에서 로봇을 구동하기 위해서는 웹 로봇 시뮬레이터와 로봇 컨트롤러를 연결하는 로봇 컨트롤 서버가 필요하다. 로봇 컨트롤 서버는 웹 로봇 시뮬레이터의 제어 명령을 로봇 컨트롤러에 전달하는 역할을 한다. 로봇 컨트롤 서버의 내부 구조를 살펴보면 그림 3과 같다. 로봇 서버는 내부적으로 크게 4개의 모듈로 구성되어진다. 클라이언트와 통신을 하기 위한 소켓 모듈, 접속한 클라이언트의 처리를 전담하고 다른 클라이언트의 접속을 위해 새로운 스레드를 생성하는 스레드 모듈, 로봇과 직렬 통신을 하는 직렬 통신 모듈, 데이터 유효성을 검증하기 위해 GUI 모듈이 있다. 클라이언트를 처리할 때 스레드를 사용하는 것은 여러 클라이언트가 접속할 경우에 서버의 부하를 덜어주기 위해서이다.

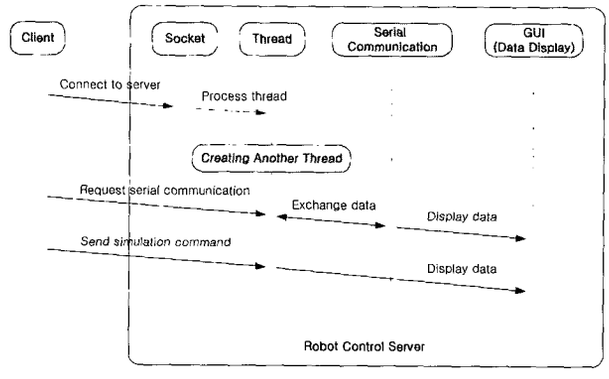


그림 3. 서버의 소프트웨어 구조.
Fig. 3. Software architecture of server.

3. 웹 기반 로봇 시뮬레이터

본 논문에서 개발한 웹 기반 시뮬레이터의 외형은 그림 4와 같다. 전체적인 웹 인터페이스는 웹 로봇의 개념에 따라서 사용하기 쉽게 제작하였다. 기존의 2차원적인 웹 인터페이스 대신에 사용자가 직관적으로 로봇의 형태를 이해할 수 있도록 3차원적으로 가시화 하였다. 웹 브라우저 내에서 실행되어지는 3차원 그래픽 어플리케이션을 작성하기 위해서 기존에는 VRML (Virtual Reality Modeling Language)나 Java3D를 주로 사용하였다 [7]-[9]. Java3D와 VRML은 플랫폼에 독립적이라는 장점이 있으나 이로 제작되어진 어플리케이션은 다른 툴로 제작되어진 어플리케이션보다 렌더링 속도가 느리다는 것과 소프트웨어적으로 재사용하기가 어렵다는 단점이 있다. 그리고 VRML은 다른

프로그래밍 언어처럼 풍부한 기능성을 제공하지 못하므로 복잡한 연산을 하기 위해서는 C++나 다른 언어를 사용해야 한다는 것과 전용 브라우저가 있어야만 사용할 수 있다는 단점이 있다[10].

본 논문에서는 VRML과 Java3D의 단점을 극복하고, 소프트웨어적으로 재사용이 가능하도록 하기 위해 Microsoft사의 COM(Component Object Model)을 사용하여 플랫폼에 독립적이고, 렌더링 속도를 향상시키는 3차원 그래픽 어플리케이션을 개발하였다. 그리고 마우스만으로 로봇의 전체적인 구조를 살펴볼 수 있도록 하여 편리하게 사용할 수 있다. 웹 기반의 시뮬레이터는 원격지상의 로봇을 제어하기 때문에 로봇을 직접 관찰하기 위하여 카메라 정보를 통해서 로봇의 움직임을 관찰할 수 있다. 웹 카메라를 사용함으로써 사용자는 원격지에서도 로봇의 위치를 확인할 수 있을 뿐만 아니라, 로봇의 움직임까지 확인할 수 있다. 카메라는 초당 한번씩 JPEG 이미지를 보내줌으로써 사용자는 원격지에서도 로봇의 운동을 확인할 수 있다. 그림 4는 시뮬레이션의 결과와 비교하기 위하여 시뮬레이터 우측에 카메라 정보창을 두어서 웹 상에서 로봇의 움직임을 관찰할 수 있도록 하였다.

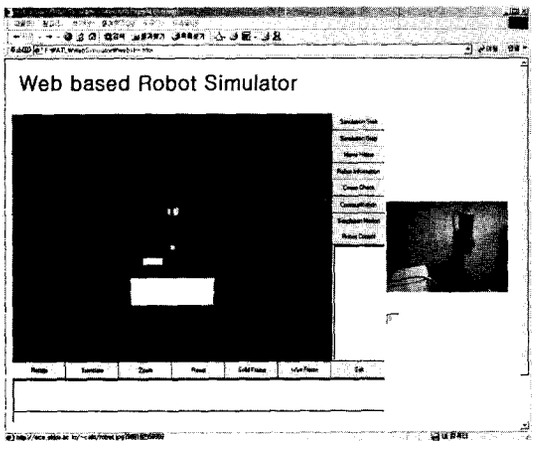


그림 4. 웹 기반 로봇 시뮬레이터 실행화면.

Fig. 4. Execution screen of web-based robot simulator.

또한, 시뮬레이터와 카메라 정보창이 한 브라우저 내에 존재하므로 온라인 시뮬레이션 결과가 제대로 수행되었는지 카메라 정보를 통해서 확인할 수 있다. 하단의 Rotate, Translate, Zoom 버튼은 가상 카메라의 위치 변경을 위한 것으로 전체적인 구성 및 시뮬레이션 과정을 좀 더 세밀하게 고찰할 수 있게 하였다. 실제 로봇을 관찰하기 위한 카메라는 고정되어 있기 때문에 시야가 한정되었는데 가상 카메라는 손쉽게 위치 변경이 가능하기 때문에 실제 카메라로는 관찰할 수 없는 부분까지도 관찰할 수 있다. 우측의 버튼은 각각 시뮬레이션 시작, 잠시 멈춤, 홈으로 리턴 및 시뮬레이션 특성을 설정할 수 있다. Robot Information버튼을 누르면 그림 5와 같이 로봇 정보 창이 나타나는데 사용자는 이를 보고 로봇의 종류 및 축 정보, 각 축의 운동반경 등을 알 수 있다. 이를 기반으로 하여 로봇의 운동반경 내에서 적절한 명령을 내릴 수 있다. 우측 하단에는 로봇의 상대적인 좌표 값이 표시되도록 하였는데 사용자는 이 값을 참고하여 로봇 끝단의 좌표 값을 직관적으로 이해할 수 있게 하였다. 그리고 하단 부분은 명령 행(Command Line)입력 창으로 실제 로봇의 터미널과 동일한 역할을 하여 실제 로봇을 제어할 때와 동일한 효과를 내도록 하였다. 실제 로봇 터미널에서도 미리 정해 놓은 태그 포인트를 이용하여 실제 명령을 내림으로써 로봇을 제어하게 된다. 예를 들어, t1이라고 하는 임의의 위치로 로봇을 이동시키기 위해서 로봇 터미널에서는 move라는 명령어를 사용하여 move t1의 명령을 내리면 t1의 위치로 이동하게 된다. 시뮬레이터에서도 이러한 기능을 추가하여 실제 로봇 터미널을 이용하는 것처럼 로봇을 제어할 수 있게 하였다. 이 부분에서는 로봇 터미널을 이용하여 가상의 로봇을 제어하는 부분이지만 로봇이 사용자의 명령에 따라 실시간으로 동작하기 때문에 실제 로봇 터미널을 이용하여 로봇을 제어하는 것과 차이가 없게 하였다. 가상 카메라의 뷰포인트를 변경하였을 때의 실행화면은 그림 6과 같다.

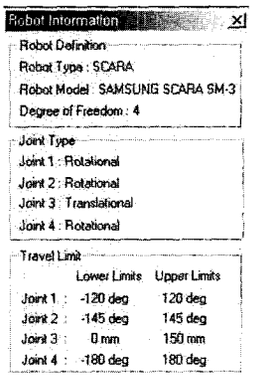
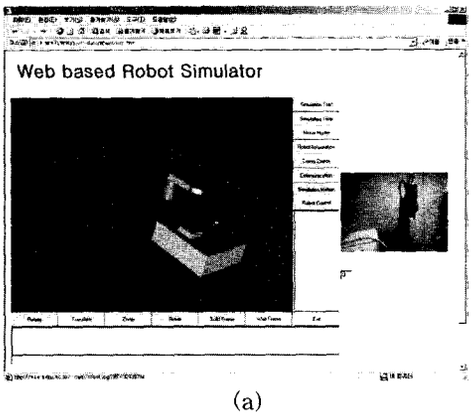
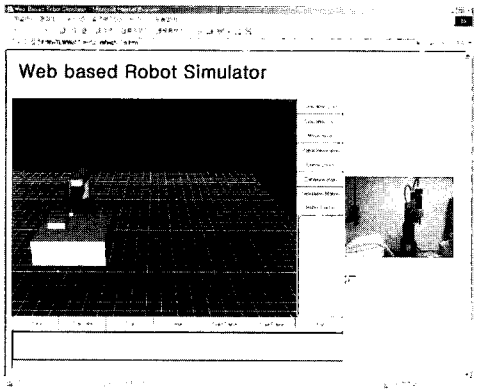


그림 5. 로봇 정보창.

Fig. 5. Robot information window.



(a)



(b)

그림 6. 가상 카메라의 뷰포인트 변경.
Fig. 6. Change viewpoint of virtual camera.

4. 로봇 컨트롤 서버

로봇 컨트롤 서버는 웹 로봇 시뮬레이터와 로봇 컨트롤러를 연결하는 역할을 한다. 로봇 컨트롤 서버의 실행화면은 그림 7과 같다. 주는 데이터와 받는 데이터를 나타내는 창은 로봇 컨트롤 서버와 로봇 컨트롤러 사이의 데이터 통신시 오가는 데이터를 표시한다. 현재 접속자수를 나타내는 창은 로봇을 제어하기 위해 로봇 컨트롤 서버에 접속한 사용자의 수를 나타내고 자세히 보기를 누르면 사용자의 이름을 확인할 수 있다. 현재는 접속자수가 없으므로 자세히 보기 버튼이 비 활성화되어 있다. 그림 8은 사용자가 로봇을 제어하기 위해 로봇 컨트롤 서버에 접속한 뒤 로봇과 통신 포트를 체크한 후에 시뮬레이션 시작 버튼을 눌러 시뮬레이션이 진행된 후의 서버의 실행화면이다. 주는 데이터는 서버에서 로봇 컨트롤러로 전송하는 데이터이고, 받는 데이터는 로봇 컨트롤러에서 서버로 오는 데이터를 의미한다. 자세히 보기 버튼을 누르면 접속한 사용자의 명단이 그림 9와 같이 나타난다. 다이얼로그 상자의 최상 단에 있는 사용자만이 로봇을 제어할 수 있게 된다.

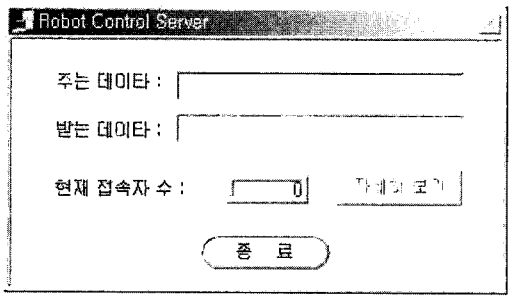


그림 7. 로봇 컨트롤 서버 실행화면.
Fig. 7. Execution screen of robot control server.

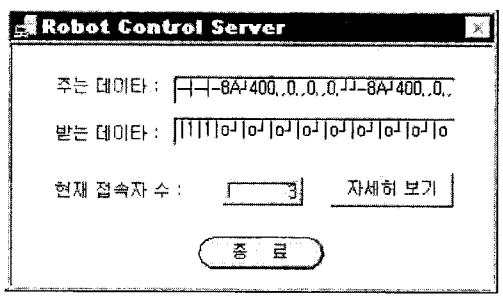


그림 8. 로봇과 컨트롤 서버간의 데이터 통신.
Fig. 8. Data communication between robot and robot control server.

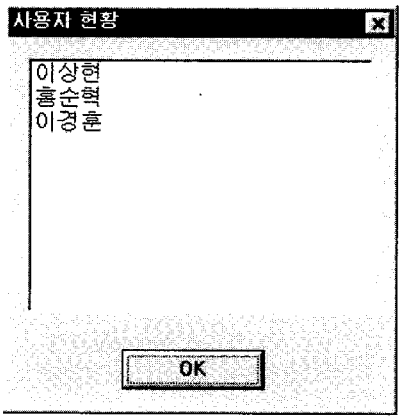


그림 9. 사용자 현황.
Fig. 9. Current user state.

5. 통신 프로토콜

로봇과 통신하기 위해서 가장 먼저 규정지어야 할 것이 통신 프로토콜이다. 시뮬레이터에서 로봇 제어 명령을 내리면 로봇 제어서버는 최종적으로 로봇에게 이를 전달하게 된다. 제어 명령이 전달되어지는 과정은 TCP/IP 프로토콜을 기반으로 하였다. 본 논문에서 사용한 클라이언트와 서버 사이의 통신 프로토콜은 그림 10과 같다. 시뮬레이터 클라이언트는 로봇 서버에 접속하기 위해 서버에 연결요청을 하게 된다. 연결

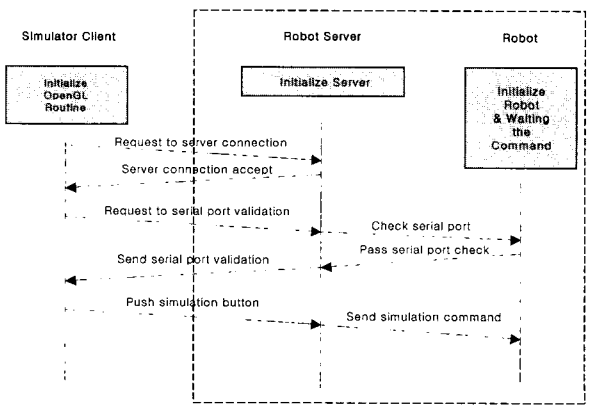


그림 10. 통신 프로토콜.
Fig. 10. Communication protocol.

요청이 수락되면 클라이언트와 서버는 정의된 패킷을 통하여 통신을 하게 된다. 클라이언트는 로봇을 제어하기 전에 로봇과 통신할 수 있는지 확인해야 한다. 로봇과 통신이 가능한지 확인하기 위해서 클라이언트는 로봇 서버에 통신 포트 유효성의 검증을 요구하게 되면, 로봇 서버는 로봇 컨트롤러와 통신 포트의 이상 유무를 체크한다. 이 결과가 다시 클라이언트에게 반영되어진다. 통신 포트에 이상이 없을 경우 클라이언트는 시뮬레이션을 수행하게 되고, 로봇에게 시뮬레이션 명령이 전달된다.

Ⅲ. 시뮬레이션

본 논문에서는 웹 상에서의 온라인 시뮬레이션을 하기 위한 중간 과정으로 오프라인 시뮬레이션을 하였다. 오프라인 시뮬레이션을 통하여 사용자는 로봇의 동작 특성 및 시뮬레이션 경로 추정 등 로봇에 대하여 다양한 정보를 얻을 수 있다. 웹 브라우저 내에서 오프라인 시뮬레이션이 진행되어지는 과정이 그림 11과 같이 나타난다. 테이블 위의 물체를 매니퓰레이트하는 작업을 수행하기 위하여 로봇 끝단에 장착된 도구를 직선으로 이동시키는 오프라인 시뮬레이션을 수행하였다. a-b-c-d 순으로 시뮬레이션이 진행되고 홈으로 돌아오는 경로는 시뮬레이션이 수행되는 반대순서이다. 가상 로봇의 이동 경로를 알아보기 쉽게 하기 위해서 가상 카메라를 약간 회전한 상태에서 시뮬레이션을 수행하였다. 오프라인 시뮬레이션은 실제 로봇에게 명령이 전달되어지는 것이 아니기 때문에 시뮬레이션이 진행되어지는 부분만을 나타냈다.

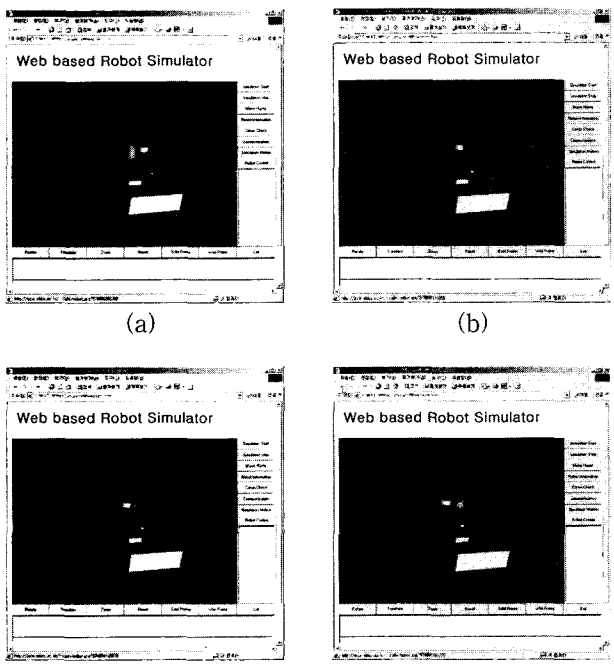


그림 11. 오프라인 시뮬레이션.
Fig. 11. Offline simulation.

로봇의 동작 특성을 알기 위해서 오프라인 시뮬레이션을 수행하고 난 다음 실제로 로봇을 제어하기 위해서 온라인 시뮬레이션을 수행하는데 오프라인 시뮬레이션과 동일한 방식으로 진행되지만 로봇과 데이터 통신을 하는 것이 차이점이다. 로봇이 항상 전원이 켜져 있는 상태가 아니기 때문에 제어를 하기 전에 먼저 통신할 수 있는지 확인을 해야 한다. 통신이 체크된 후에 사용자는 로봇 프로그램을 통하여 로봇을 제어할 수 있다. 오프라인 시뮬레이션과 동일한 방법으로 시뮬레이션 시작 버튼을 누르면 시뮬레이션이 시작되게 된다. 시뮬레이터의 로봇의 움직임에 따라 실제 로봇도 동일하게 움직이는지 확인하기 위해서 웹 카메라를 이용하여 로봇의 움직임을 살펴볼 수 있다. 온라인 시뮬레이션은 사용자의 제어 명령의 결과를 실제로 보여주어야 하는데 이는 카메라 정보를 통하여 알 수 있다. 온라인 시뮬레이션이 진행되어지는 과정은 그림 12와 같다. 온라인 시뮬레이션도 오프라인 시뮬레이션과 동일하게 진행되어지므로 시뮬레이터가 변경되는 모습보다도 사용자의 실제 명령이 제대로 로봇에게 전달되는지 확인하기 위해 카메라 정보창을 통하여 관찰된 로봇의 움직임만을 나타냈다.

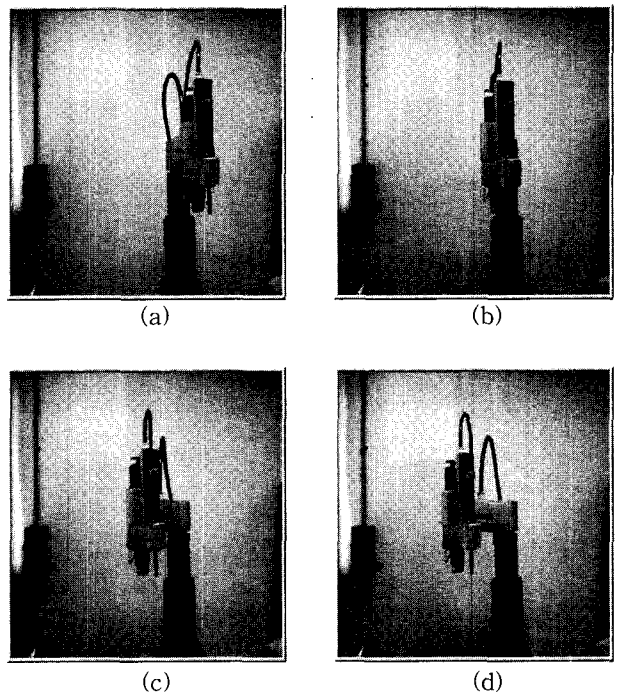


그림 12. 카메라 정보창을 통한 온라인 시뮬레이션 확인.
Fig. 12. Online simulation verification by camera information window

카메라 정보를 통하여 로봇의 이동경로와 운동을 관찰할 수 있는데, 이 카메라는 고정되어 있기 때문에 여러 각도에서 로봇의 운동을 관찰하기에는 문제점이 존재한다. 그러나 본 논문에서 제안한 시뮬레이터를

사용함으로써 가상 카메라의 각도를 변화시킬 수가 있기 때문에 로봇의 움직임을 여러 각도에서 살펴볼 수가 있다. 카메라 정보를 바탕으로 로봇의 움직임을 살펴볼 수 있고, 가상 카메라의 각도를 시뮬레이션 중간에 변경함으로써 현재 로봇의 움직임을 여러 각도에서 관찰할 수 있다. 온라인 시뮬레이션이 진행되는 도중에 가상 카메라의 각도를 변경하여 로봇의 움직임을 관찰한 것은 그림 13과 같이 나타난다. 그림 13은 가상 로봇의 위쪽에서 가상 카메라를 이용하여 살펴본 것을 나타낸다. 카메라 정보창에는 실제 카메라의 이미지를 전송하게 되는데 그림에서 보는 것처럼 실제 카메라 이미지는 고정되어 있을 뿐만 아니라 줌기능이 없기 때문에 시뮬레이션을 진행할 때 정해진 위치에서만 로봇을 고찰할 수 있지만, 가상카메라를 사용하면 사용자가 관찰하고 싶은 로봇의 여러 부분들을 살펴볼 수가 있다. 즉 가상 카메라는 실제 카메라로 고찰할 수 없는 부분을 가상 로봇을 통하여 살펴봄으로써 로봇의 동작 특성 및 여러 움직임에 대하여 알 수 있도록 하였다.

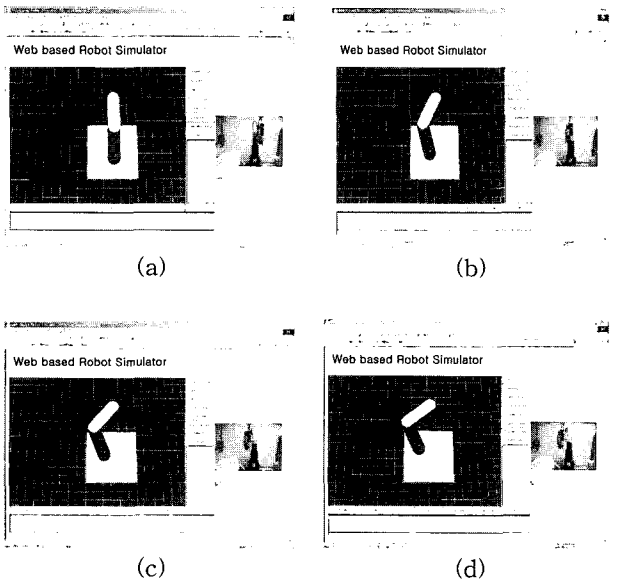


그림 13. 가상 카메라를 변경하여 관찰한 온라인 시뮬레이션.

Fig. 13. Online simulation observed by changing the virtual camera.

Ⅳ. 결론

본 논문은 기존 시뮬레이터의 한계점인 사용자의 위치 종속적인 것과 웹에서 시도되어진 시뮬레이터가 존재하지 않는 것을 고려하여 웹 브라우저 내에서 실행되어지는 3D 로봇 시뮬레이터를 개발하였다. 이는 최근 인터넷의 급속한 발전으로 웹에 접속할 수 있는 공간의 확대로 인하여 시뮬레이터를 사용하는 사용자의 위치 투명성을 제공하는 장점이 있다. 그리고 웹 로봇의 개념에 따라 사용자가 사용하기 쉽도록 인터페이스를 구성하였을 뿐만 아니라 로봇의 동작특성을 충분히 이해하도록 하였다. 원격지에서 로봇을 제어하는 것이기 때문에 로봇의 움직임을 살펴볼 수 있도록 카메라 이미지를 웹 상에서 확인하도록 하였다. 현재는 주기적으로 JPEG 이미지를 전송하는 방식을 취하였는데, 네트웍 사정으로 인하여 이미지의 일부분만이 보이거나 몇 초에 한번씩 이미지가 업로드되어, 실시간으로 로봇의 움직임을 고찰하는 부분에는 한계점이 존재하였다. 시뮬레이터에 더욱 더 많은 기능 추가와 JPEG 이미지 대신에 MPEG을 전송함으로 실시간으로 로봇을 고찰하는 것이 향후 연구과제이다.

참고문헌

[1] Ken Goldberg, Steven Gentner, Carl Sutter, and Jeff Wiegley, "The mercury project : A feasibility study for Internet Robots," *IEEE Robotics & Automation Magazine*, pp. 35-40, March, 2000.

[2] Patrick Saucy and Francesco Mondada, "KhepOnTheWeb : Open access to a mobile robot on the internet," *IEEE Robotics & Automation Magazine*, pp. 41-47, March, 2000.

[3] Dirk Schulz, Wolfram Burgard, Dieter Fox, Sebastian Thrun, and Armin B. Cremers, "Web interfaces for mobile robots in public places," *IEEE Robotics & Automation Magazine*, pp. 48-56, March, 2000.

[4] Hirohisa Hirukawa, Toshihiro Matsui, and Hiromu Onda, "Prototypes of teleoperation systems via a standard protocol with a standard human interface," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, vol 2, pp. 1028-1033, April, 1997.

[5] 윤병준, 이종수, 최경삼, "Web을 이용한 로봇 매니퓰레이터의 원격제어," *Pro. of the 14th KACC*, pp. A-441-444, October, 1999.

[6] 정광식, 서석환, 서윤호, "Web 기반 가상공작기계 모델링 및 구현," 한국 CAD/CAM 학회 학술발표회 논문집, pp. 1-6, 2000.

[7] Yuhua Luo, Ricardo Galli, Miguel Mascaro, and Pere Palmer, "Cooperative design for 3D virtual scenes," *Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems*, pp. 373-383, 2000.

[8] Tain-chi Lu, Chuanwen Chiang, Ming-tang Lin, and Chungnan Lee, "A collaborative scene editor for VRML worlds," *Proceedings of the Eurographics'98*, pp. 53-61, 1998.

[9] Josue Jr. G. Ramos, Silvio M. Maeta, Marcel Bergerman, Smuel S. Bueno, Luiz G. B Mirisola,

and Augusto Bruciapaglia, "Development of a VRML/JAVA unmanned airship simulating environment," *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 1354-1359, 1999.

[10] Werner Geyer and Martin Mauve, "Integrating support for collaboration-unaware VRML models into cooperative applications," *Proceedings of the IEEE Multimedia Systems '99*, pp. 655-660, 1999.

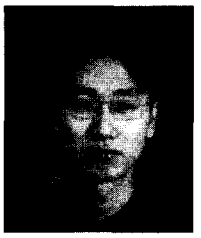
[11] 홍순혁, 전재욱, 윤지섭, "실시간 로봇 시뮬레이터 개발," *Proc. of the 14th KACC*, pp. A-409-412, October, 1999.

홍 순 혁

1998년 성균관대학교 제어계측공학과 졸업. 2000년 성균관대학교 전기전자 및 컴퓨터 공학부 석사졸업. 2000년~현재 성균관대학교 전기전자 및 컴퓨터 공학부 박사과정 재학중. 관심분야는 인터넷 기반 로봇, 가상현실 시뮬레이션, 원격조작, 실시간 내장시스템.

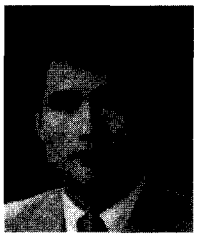
전 재 욱

1984년 서울대 공대 전자공학과 졸업. 1986년 동 대학원 전자공학과 졸업(석사). 1990년 Purdue University, Ph.D. 1990년~1994년 삼성전자 생산기술센터 선임연구원. 1994년 성균관대학교 제어계측공학과 조교수. 현재 성균관대학교 전기전자 및 컴퓨터 공학부 부교수. 관심분야는 로봇공학, 비주얼서보잉, 마이크로프로세서.



이 상 현

1998년 성균관대학교 전기공학과 졸업. 1998년~현재 성균관대학교 전기전자 및 컴퓨터 공학부 석사과정 재학중. 관심분야는 웹 기반 로봇, 가상현실, 원격조작.



윤 지 섭

1980년 서울대학교 기계공학과 학사. 1987년 한국과학기술원 기계공학과 석·박사. 1987년~현재 한국원자력연구소 핵연료주기단 원격취급장치개발실 실장/책임연구원. 관심분야는 수중로봇기술, 생산자동화 및 계측제어, 극한환경 원격취급기술.