

〈논문〉 SAE NO. 2000-03-0077

자동차 시뮬레이터의 가상환경 구성에 대한 연구

Construction of Virtual Environment for a Vehicle Simulator

장재원*, 손권**, 최경현***
Cheawon Chang, Kwon Son, Kyunghyun Choi

ABSTRACT

Vehicle driving simulators can provide engineers with benefits on the development and modification of vehicle models. One of the most important factors to realistic simulations is the fidelity given by a motion system and a real-time visual image generation system. Virtual reality technology has been widely used to achieve high fidelity. In this paper, the virtual environment including a visual system like a head-mounted display is developed for a vehicle driving simulator system by employing the virtual reality technique. Virtual vehicle and environment models are constructed using the object-oriented analysis and design approach. According to the object model, a three dimensional graphic model is developed with CAD tools such as Rhino and Pro/E. For the real-time image generation, the optimized IRIS Performer 3D graphics library is embedded with the multi-thread methodology. Compared with the single loop approach, the proposed methodology yields an acceptable image generation speed, 20 frames/sec, for the simulator.

주요기술용어 : Vehicle Simulator(차량 시뮬레이터), Virtual Reality(가상 현실), Virtual Environment(가상 환경), Visual System(가시화 시스템), Virtual Prototype(가상 원형)

1. 서론

차량 시뮬레이터는 운전자가 실제 차량을 조작하는 것과 같은 느낌을 제공하는 장치이다. 이를

통해 운전자가 차량을 조작하는 동안의 행위와 차량의 특성을 분석하고, 이로 인해 발생하는 차량의 운동을 해석하여, 그 결과를 운동시스템 및 시청각 시스템을 통해 운전자에게 정보를 제공하여 실제로 차량을 운전하는 것처럼 느끼게 한다.

차량 시뮬레이터는 차량의 설계 및 개발 단계에서 중요한 역할을 담당하고 있다. 동시공학 개념의 실제적 적용을 통해 차량 개발 기간을 단축하여 시간과 비용의 절감효과를 얻기 위해서는

* 회원, 부산대학교 대학원

** 회원, 부산대학교 기계공학부

*** 회원, 제주대학교 기계에너지생산공학부

각 설계 데이터를 가상 원형(virtual prototype)으로 검증하고, 자동차 시뮬레이터에 의한 시험을 행하여 설계단계에서 이루어지던 많은 반복 수정 작업을 줄이는 노력이 필요하다. 선진국의 경우 벤츠자동차^{1,2)}와 아이오아대학^{3,4)} 등 자동차 회사와 대학을 중심으로 차량 시뮬레이터를 개발하여 자동차의 설계 및 시험에 이용함으로써 많은 시간과 비용을 절약하고 있다. 국내에서도 대학, 기업체 연구소 등에서 항공기 및 중장비 시뮬레이터^{5,6)}를 개발하였으며, 최근에는 자동차 시뮬레이터^{7,8)}의 개발에 대한 연구가 진행되고 있다.

차량 시뮬레이터는 운전자에게 많은 시각적 정보를 공급하기 때문에 현실감 있는 시각정보의 제공이 중요하다. 일반적으로 시각정보를 제공하기 위해 워크스테이션과 3차원 그래픽 도구를 사용하여 이미지를 생성하고, 생성된 이미지를 프로젝터 등의 표시장치를 통해 운전자에게 전달하게 된다. 이 과정에서 현실감 있는 시각정보를 실시간으로 처리하여야 하므로 시각정보의 생성 및 전달 과정은 상당히 중요하다. 운전자 입력에 대응하는 효과적인 시각정보의 생성과 현실감 있는 이미지 생성을 위해서 최근에는 가상현실 기술을 적극 도입하여 이용하고 있다.⁹⁾

본 논문에서는 운전자에게 몰입감을 제공하는 가상현실 기술을 적용하여 실시간 자동차 시뮬레이터를 위한 효율적인 가상환경 및 시각시스템 개발에 관하여 기술하였다. 가상 환경과 차량을 OOAD(object-oriented analysis and design) 기법을 이용하여 객체로 모델링하고 이들 객체 사이의 관계를 정의하였다. 모델링 된 객체를 3차원 모델링 도구와 3차원 그래픽 라이브러리 및 디스플레이 시스템을 활용하여 효율적으로 가시화 하는 방법을 제시하였다. 가상현실 입력장치를 사용하여 운전자의 입력을 처리하고, 실제 환경을 모사한 주행 환경을 현실감 있게 제시하는 그래픽 시뮬레이터구성에 대하여 기술하였다.

2. 그래픽 시뮬레이터 시스템의 구성

전체 시스템은 Fig. 1과 같이 크게 워크스테이션에서 처리되는 소프트웨어와 가상현실 장비를 포함한 하드웨어로 나눌 수 있다. 워크스테이션은 전체 시스템을 관장하는 역할과 그래픽 라이브러리를 이용하여 3차원 이미지를 생성하는 부분을 담당한다. 가상현실 장비는 운전자의 조작을 입력받아 워크스테이션으로 전달해 주며, 또한 생성된 이미지 출력을 헤드 마운티드 디스플레이(head-mounted display, HMD)를 통해 운전자에게 전달해 주는 역할을 한다.

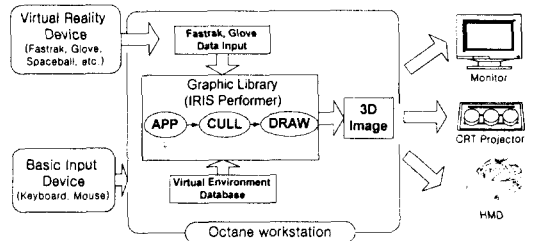


Fig. 1 Structure of graphic simulator system

2.1 소프트웨어 구성

자동차 시뮬레이터를 위한 가상현실 소프트웨어는 3차원 그래픽 제시, 가상현실 장치 인터페이스, 통신 및 사용자 입력 처리 등의 기능을 실시간으로 구현해야 한다. 현실감 있는 자동차 및 주행 환경 이미지를 생성하고, 가상현실 장치들을 통해 탑승자와 상호작용을 하며, 동역학 해석기 및 운동감 제시기와 실시간으로 정보 전달이 가능해야 한다.

3차원 가상 그래픽 환경을 구성하기 위해 그래픽 라이브러리로 실리콘 그래픽스사의 아이리스 퍼포머(IRIS Performer)^{10,11)}를 사용하였다. 퍼포머는 IRIS GL을 기본으로 만들어졌으나 실리콘 그래픽스사의 하드웨어에 최적화된 알고리즘을 사용하여 폴리곤의 실시간 처리를 위한 함수들을 지원한다. 따라서 실시간 그래픽 처리가 필요한 시뮬레이션 소프트웨어 개발에 우수한 환경을 제공하며, 다양한 3차원 CAD 데이터를 이용할 수

있는 데이터 읽기 관련 함수를 지원하고 있다.

모델링을 통해 구성된 가상환경 데이터베이스로부터 3차원 이미지를 생성하기 위해서는 어플리케이션 트래버설(application traversal, APP), 컬 트래버설(cull traversal, CULL), 드로우 트래버설(draw traversal, DRAW)의 3가지 기본 과정을 거치게 된다. APP과정은 가상환경내의 다양한 변환들을 계산하는 과정이다. CULL과정은 가려지거나 보이지 않아 그려질 필요가 없는 정보들을 제거하여 렌더링 속도를 향상시키는 계산과정이다. DRAW과정은 렌더링 및 이미지 생성 명령들을 수행시킨다. 이렇게 생성된 3차원 이미지가 프레임 버퍼에 저장되고 이전 이미지를 갱신할 때 표시장치를 통해 제공된다.

가상현실 인터페이스 및 통신은 시리얼 포트 또는 랜(LAN)을 통해 이루어지며, 사용자 입력의 처리를 위해 편리한 GUI환경을 제공한다. 가상현실 인터페이스에서의 시간지연을 줄이기 위해 병렬 처리를 도입하고, 엑스 윈도우(X-window) 환경에서의 GUI 설계를 위해 모티프(Motif)라이브러리를 사용한다.

2.2 하드웨어 구성

가상현실 장비를 포함한 그래픽 시뮬레이터의 하드웨어 인터페이스의 구성은 Fig. 2와 같다. 자동차 시뮬레이터를 위한 동역학 해석기 및 운동재현기와는 허브를 통해 연결되며 TCP/IP 프로토콜을 이용하여 정보를 전달한다. 본 연구에서 그래픽 시뮬레이터 하드웨어는 데이터글러브(data glove), 공간 센서, HMD 등의 출력 장치를 포함한 워크스테이션으로 구성된다.

패스트랙(Fastrak)¹²⁾은 공간상의 위치와 방위를 측정하는 6자유도 센서로 탑승자의 머리와 손에 부착하여 시점의 변화와 손의 위치 및 방위를 검출한다. 패스트랙은 전자기파를 방출하는 트랜스미터로부터의 상대적인 거리 및 방위를 측정하는 센서이며, 탑승자의 이동 범위 이내에서는 일반적인 실험실 환경에서 발생하는 노이즈에 의한

오차가 가시적 범위를 넘지 않는 결과를 제공한다. 패스트랙의 동작 지연은 4 ms이며, 응답대기에 의한 지연현상을 병렬 처리를 통해 제거하였다.

데이터글러브로 사용된 5th-Glove¹³⁾는 5개의 광섬유를 사용하여 각 손가락의 굽힘량을 검출한다. 자체에 2자유도 틸트(tilt) 센서를 가지고 있으나 패스트랙을 사용한 위치 방위값을 사용하므로 틸트 센서의 값은 사용하지 않는다. 5th-Glove는 5자유도 센서로 손가락의 각 마디의 움직임을 얻을 수 없으며, 비례 기준값에 대한 상대적인 보정이 요구된다. 본 연구에서는 초기 상태에서 최대 굽힘과 완전 이완으로부터 얻어진 값을 기준으로 각 손가락에 대해 비례적인 관절 각도 변화를 유발시켜 가상환경에서 운전자의 손 동작을 표현한다.

헤드 마운티드 디스플레이는 다른 디스플레이 장치들과 달리 트랙커(tracker)를 부착할 경우 전방위의 표현이 가능하고, 사용자에게 몰입감을 제공해 주는 장점이 있다. 그러나 일반적으로 컴퓨터 모니터보다는 해상도가 높지 않으며 또한 안정성, 착용감 등에서 사용자에게 거부감을 줄 수도 있다. 본 연구에서는 Virtual Research System사의 V6 HMD¹⁴⁾를 사용하였다. V6는 2개의 1.3 인치 LCD를 장착한 반 헬멧형상의 시각정보 출력 장비로 해상도는 640×480이다. 스크린과 프로젝터를 사용하는 시스템이 시점변화에 대응할 수 없는 단점을 보완하여 효율적으로 시각정보를 제공해 준다.

워크스테이션이 가지는 그래픽 출력 장치는 NTSC신호를 얻을 수 있는 PVB(personal video board)와 여러 개의 RGB신호를 출력할 수 있는 다중 채널 옵션¹⁵⁾을 가진다. 이를 통해 모니터 출력, 프로젝터를 통한 스크린 투사, 또는 HMD를 통한 스테레오 이미지 제공이 가능하다.

가상현실 입출력 장비들은 모두 워크스테이션과 RS-232C 시리얼 통신을 통해 데이터를 주고받는다. 각 장비에서 측정된 데이터는 실제계와 가상환경 등에서 정의되는 좌표계와 일치하지 않

으므로 좌표축 변환 및 스케일 조절을 통해 측정값을 가상 공간에서의 좌표축과 일치시켜 사용된다.

구성된 가상현실 인터페이스는 일반적인 실험실 환경에서 트랜스미터와의 거리가 1 m 이내일 때 운전자 시점의 변화에 대응시킨 결과, 패스트랙은 $\pm 60^\circ$ 의 전방시야에 대해 안구 운동을 고려하지 않을 경우 약 0.7° 이내의 오차로 시점 변화를 반영 할 수 있다. 또한 시스템은 독립된 2개의 채널로 구성되어 모노 뷰(mono view)와 스테레오 뷰를 지원한다. 패스트랙의 세동과 노이즈의 영향을 줄이기 위해 미분을 통한 검사와 평활 과정을 포함하고 있다. 본 연구에서 가상환경의 출력은 모니터, 프로젝터, 그리고 HMD에 의해 제공되며, 장시간의 주행과 관련한 HMD 착용자의 불편감과 부작용에 관련된 실험적인 평가는 배제하였다.

청각과 촉각 등 다른 환경도 실제와 유사하게 제공되어야 한다. 본 연구에서는 전체 가상현실의 환경을 이루는 여러 가지 요소 중에서 시각 부분을 대상으로 보다 우수한 시각 정보 제시 능력을 구현하기 위한 방법을 제시하고자 하였다.

시각적인 환경은 크게 차량 모델과 가상환경 모델로 구성된다. 차량 모델은 시뮬레이터에서 운전자가 조작하게 되는 차량과 주변 차량들이고, 가상환경은 주행 중에 나타나는 주변의 도로 및 건물 등의 환경을 의미한다. 이러한 차량과 가상환경을 객체지향 모델링 기법의 하나인 OOAD¹⁶⁾ 표기법을 이용하여, 객체를 모델링하고 객체들 사이의 관계를 정의하였다. 그리고 모델링 된 객체는 그래픽 라이브러리와 3차원 모델링 도구를 이용하여 그래픽 모델을 생성하여 가상환경에 적용하였다.

3.1 환경의 객체 모델과 그래픽 모델

객체 모델은 시스템의 객체와 객체 사이의 관계, 각 객체의 속성과 기능으로 시스템의 정적구조를 정의한다. 여기서는 객체 모델의 기술방식 중의 하나인 Booch의 방법론을 사용하여 객체 모델을 실제 차량의 특성에 근거하여 정의하였다. 각 객체에 대한 특성을 클래스를 사용하여 객체 속성과 동작에 따라 관계를 정의하였다.

클래스란 공통적인 구조와 행위를 공유하는 객체의 집합이며, 객체와 클래스는 다른 객체 또는 클래스와 연관관계, 상속관계, 소유관계, 사용관계 등의 관련성을 가지고 있다. 연관관계는 객체가 논리적인 연결관계를 가지는 경우를 말하며, 상속관계는 하나의 클래스와 그 클래스로부터 나누어진 다른 클래스와의 관계를 말한다. 소유관계는 물리적으로 소유 또는 부분을 이루는 경우를 말하며, 사용관계는 다른 클래스를 사용하는 경우를 나타낸다. 객체로 모델링 된 차량과 가상환경을 실제 차량 시뮬레이터에 사용하기 위해서는 그래픽 모델 생성 과정이 필요하다. 그래픽 모델을 생성하기 위해 범용 3차원 그래픽 모델링

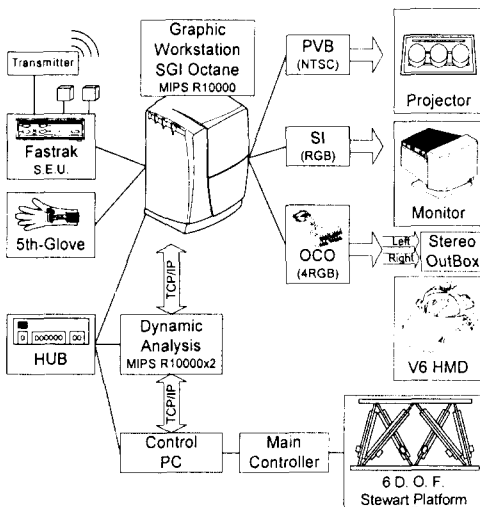


Fig. 2 System hardware interface

3. 가상환경 및 차량의 모델링

일반적으로 가상현실의 환경은 사용자가 실제와 같은 느낌을 받을 수 있도록 제작되어야 한다. 그러기 위해서는 시각적인 이미지 뿐 만 아니라

도구인 라이노(Rhino 3D)와 CAD 프로그램인 프로 엔지니어(Pro/ENGINEER)를 사용하였다. 그래픽 파일 포맷으로의 변환을 위해서 사용이 쉽고 보편적으로 사용되는 CAD 툴에서 편집이 가능한 3ds 메쉬 포맷을 선정하고, Fig. 3과 같이 파일 포맷 변환과 텍스처 맵핑의 과정을 통해 가상 환경에서 사용되는 그래픽 객체를 생성하였다.

가상환경은 건물과 도로, 자연 환경, 각종 표지판, 기타 주변 장치물 등으로 구성된다. 환경과 건물객체의 연산요소를 줄이고 실제와 같은 현실감을 부여하기 위해 텍스처 맵핑 기법을 사용하였다. 디지털 카메라를 이용한 실사 이미지를 텍스처 소스로 사용하면 환경 객체의 사실성을 증가시킬 수 있다. 따라서 환경 모델에 사용되는 폴리곤 수를 대폭 감소시킬 수 있어 실시간 렌더링 과정에서 하드웨어에 걸리는 부하를 감소시키는 효과를 가져올 수 있다. 가상 주행환경의 대상이 되는 실세계의 환경 객체로부터 건물 이미지를 촬영하여 이미지 편집과정을 통해 재구성한다. 실세계의 환경 객체에 대한 스케일 정보를 얻어, 이를 기초로 3차원 형상을 생성하고, 편집된 이미지를 텍스처 좌표 설정에 따라 맵핑한다. 이 과정을 거치면 환경 모델의 복잡성을 줄이고, 현실감을 증대시킬 수 있어, 3차원 가상환경 이미지의 갱신 시간을 단축시키는 효과를 얻을 수 있다.

가상환경에 대한 객체 모델은 Fig. 4와 같다. 가상환경은 크게 도로, 각종 표지, 조명, 건물과 자연환경으로 구성된다. 도로는 형태에 따라 비포장도로와 포장도로로 구분되며 포장도로는 중앙선, 옆선, 도로 표지판, 신호등 등을 가진다. 표지는 표지판과 신호등으로 나눌 수 있으며 상속관계로 정의하였다. 조명은 가로등, 태양으로 나뉘지며 신호등과 함께 조명과 상속관계로 정의하였고, 건물은 각종 집, 아파트, 빌딩과 역시 상속관계로 정의하였다. 자연환경은 하늘, 나무, 산, 강과 소유관계로 정의하였고, 하늘은 태양과 구름, 안개와 역시 소유관계로 정의하였다.

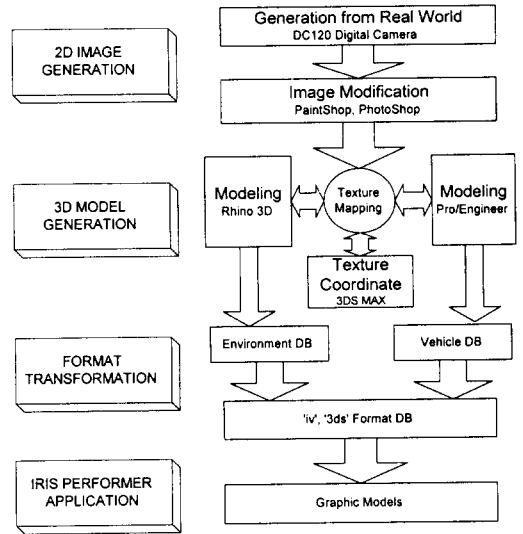


Fig. 3 Graphic modeling process

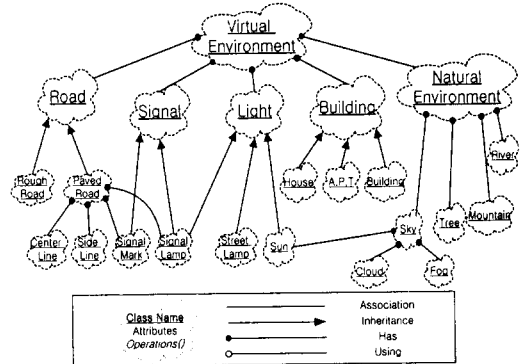


Fig. 4 Object model of virtual environment

환경의 그래픽 모델은 형상, 가시화와 관련된 재질 특성, 그리고 텍스처 좌표 정보를 가진다. 해석기로 전달되는 도로의 데이터에는 형상정보와 함께 범프와 같은 노면 장애물에 대한 위치도 포함되며, LAN을 통해 시작전에 전달된다. 해석기에서 노면 특성은 건조한 아스팔트 도로의 마찰계수 $0.85 \mu p$ 를 기본값으로 하고 있다.

Fig. 5는 건물의 그래픽 모델 생성시 텍스처 맵핑 기법을 구현하는 과정을 간략히 나타내었다. 실제 건물 객체가 가지는 복잡성을 최외곽면의 모델링과 실사에 의한 텍스처 맵핑 기법을 사용하여 처리시켰다. 이로 인해 환경 객체의 표현

에서 사실성 향상 및 실시간 렌더링을 위한 계산량 감소라는 두가지 부분을 동시에 만족시킬 수 있다

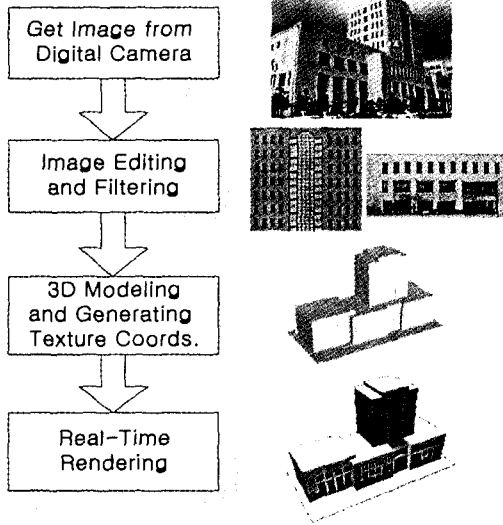


Fig. 5 Building model process with texture mapping technique

3.2 차량의 객체 모델과 그래픽 모델

차량의 객체 모델은 Fig. 6과 같다. 차량은 프레임틀을 기준으로 각종 장치가 프레임틀에 속해 있다. 따라서 프레임틀은 의자, 범퍼, 후드, 트렁크, 문, 라이트 시스템, 엔진, 파워트레인 시스템, 브레이크 시스템, 스티어링 시스템, 서스펜션 등을 소유하는 관례로 정의된다. 라이트 시스템, 엔진, 파워트레인 시스템, 브레이크 시스템, 스티어링 시스템 등은 각 객체에 영향을 미치는 스위치, 페달 등과 논리적 연결관계로 정의된다. 스위치와 페달 등은 운전자가 조작하는 장치들이므로 운전자와 사용관계로 정의하였다. 휠은 파워트레인 시스템, 스티어링 시스템, 브레이크 시스템, 서스펜션과는 논리적 연결관계로, 또한 타이어와는 소유관계로 정의하였다. 타이어는 직접 접하게 되는 도로와 연관관계에 해당하며, 논리적 연결관계로 정의하였다.

최근 차량 개발 과정에서 가상 원형을 통한 설계 검증 기술이 활발히 도입되고 있다. 부품의 설계 데이터들을 조립하여 가상원형을 생성함으로써 물리적인 원형을 생성하기 이전에 차량의 형상정보와 물리정보를 얻을 수 있다. 따라서 차량 시뮬레이터는 가상원형을 수행시험 할 수 있는

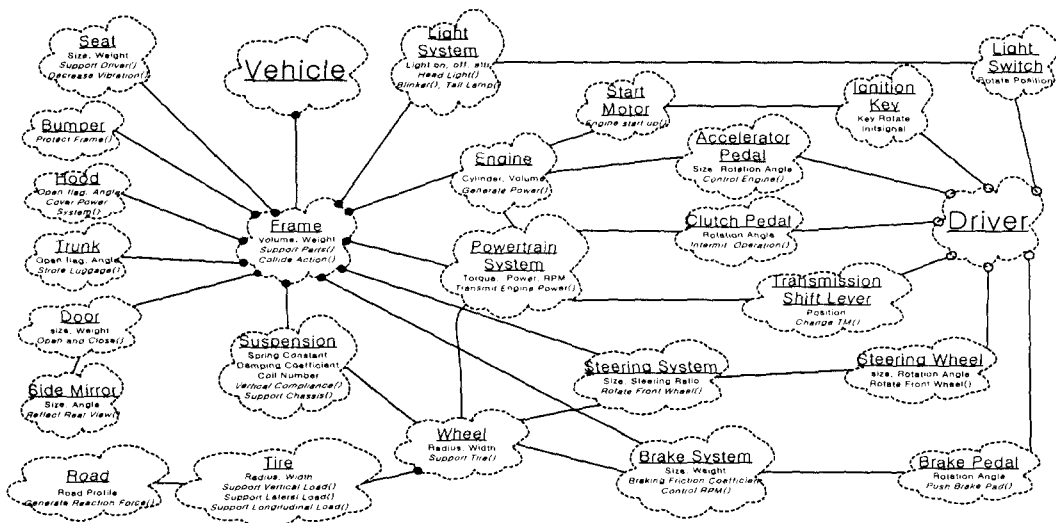


Fig. 6 Object model of virtual vehicle

도구로써 활용될 수 있다. 이러한 가상원형 생성 과정을 모의하여 각 부품 객체의 모델링을 수행하고 차량객체 계층구조에 따라 조립되도록 설계하였다. 정밀한 모델링과 계층 구조 생성 및 해석에 필요한 물리적 특성 생성 등의 강력한 설계 능력을 보유한 프로 엔지니어를 이용하였다. 차량 부품 객체를 설계하고, 각 부품을 가상 조립하여, 완성 모델을 생성하였다. 이를 가상환경에서 사용 가능한 메쉬 파일 포맷, 즉 'iv', '3ds', 'IGES', 'STL' 등으로 변환하였다. Fig. 7은 생성된 차량의 부품 객체 및 조립 모델이다.

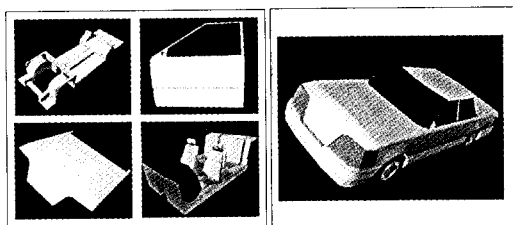


Fig. 7 Vehicle model

4. 그래픽 시뮬레이터

Fig. 8은 본 연구를 통해 개발된 가상현실 주행 시뮬레이터 소프트웨어(virtual reality driving simulator, VDS)의 동작 화면을 보여준다. VDS는 C/C++언어를 사용하여 UNIX를 기반으로 하는 X-WINDOW환경에서 개발하였으며, OpenGL을 지원하는 UNIX시스템에서 동작될 수 있다. 모티프를 사용하여 편리한 사용자 인터페이스는 가상현실 장비의 설정기능, 그래픽 환경 설정, 모델 객체 관리기능, 그리고 다양한 비디오 신호 출력 포맷을 지원한다. 초기화 및 환경 설정 후 데이터 베이스를 불러들여, VDS의 기동이 이루어지면 주행 시뮬레이션을 시작할 수 있다. 결과의 저장 및 후처리기능을 수행하고 종료되는 순서로 동작한다. 그래픽 시뮬레이터의 실시간 동작을 위해서 모델의 복잡도를 조절하고 병렬 처리를 도입함으로써 요구되는 갱신율을 충족시킬 수 있다.

4.1 모델 복잡도 조절

실시간 차량 시뮬레이터의 그래픽 환경은 시간 지연이 없어야 한다. 따라서 일정 수준 이상의 갱신율을 유지해야 하며, 일반적으로 20 프레임/초 이상의 갱신율이 요구된다. 이러한 조건을 만족시키기 위해 LOD(level of detail) 기능, 배면 삭

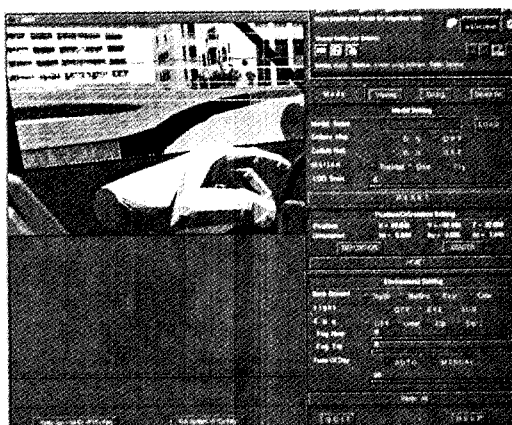


Fig. 8 Monitor screen in graphic simulator

제, 텍스처 맵핑(texture mapping)처리 등의 기법들이 적용된다. 또한, 그래픽으로 구현될 환경을 미리 데이터베이스로 구축하여 이미지 생성을 위한 초기화 시간을 단축시킨다.

3차원 이미지를 구현하기 위해서는 많은 계산이 필요하다. 실시간으로 처리하기 위해서는 만들어진 환경의 데이터를 과도한 사실성의 손실이 없는 범위에서 하드웨어의 처리능력을 만족시키는 수준으로 최소화하여야 한다. 또한 가상현실 장비, 동역학 및 운동감 시스템과의 연결시에도 적절한 수준의 화면 갱신율을 유지하는 것이 상당히 중요하다. 그러므로 가상환경을 구성하기 위해서는 하드웨어 사양에서 이러한 부분을 만족할 수 있도록 모델을 재처리하여 가시화해야 한다.

운전자 시점은 운전자 정보에 의해 구성된 형상 치수에 의해 초기위치가 결정되며, 패스트랙의 입력값에 의해 위치가 실시간으로 수정된다. 또한, 마우스 입력을 통해 가상환경을 임의로 둘

리 볼 수 있도록 자유 시점 배치기능을 포함하고 있다. Fig. 9는 모델링된 가상환경 정보를 C/C++ 언어로 제작한 가상현실 주행 시뮬레이터 프로그램에서 2가지 시점으로 나누어 가시화 한 예이다.

Table 1은 Fig. 9와 같이 구성된 가상환경 정보와 가상원형 및 주행 시뮬레이션을 위해 생성된 차량모델에 대한 정보를 나타낸다. 실시간 렌더링의 화면 갱신을 하한으로 20.0 프레임/초를 평가기준으로 하였을 때 환경모델은 만족하였으나, 프로 엔지니어로부터 생성된 차량모델의 메쉬는 실시간성을 충족시키지 못한다. Fig. 10은 단일 시야 질두체(view frustum)내에서 약 23,000개 이하의 삼각면으로 구성된 모델로 20프레임 이상의 화면 갱신을 얻을 수 있음을 보여준다.

차량모델은 곡면과 작은 부품에 메쉬가 집중되어 있다. 따라서 메쉬를 생성할 때 가상 설계 검증에 사용하기 위한 것과 실시간 주행 시뮬레이션을 위한 것으로 분리 생성할 필요가 있다. 하드웨어의 처리 능력을 고려한 수준으로의 조절과정을 거쳐야 효율적으로 객체의 표현이 가능하다. 프로 엔지니어의 자동 메쉬 생성 인자를 사용하는 경우 곡면에 대하여 대단히 근사한 메쉬를 생성하도록 설정되어 있다. 이는 프로 엔지니어내부에서 곡면 정보에 대해 실제적 계산을 행하도록 처리되며, 메쉬 생성시 이러한 정보의 손실이 크지 않도록 메쉬를 구성하기 때문이다. 이로 인해 280 메가바이트 이상의 용량과 80만개 이상의 삼각형 메쉬가 생성되었다. 이러한 대용량 메쉬를 실시간 렌더링 하기 위해서 요구되는 하드웨어의 성능은 현실적인 하드웨어의 범위를 훨씬 초과한다. 메쉬 생성에 관한 2가지의 곡면에 대한 수학적 보간 인자인 현의 높이와 각 제어(angle control) 인자를 조절하여 차량의 삼각형 메쉬수를 상당량 감소시킬 수 있다. 실제 생성 결과에서 10만개 이상에서부터 2만개 정도의 삼각형 메쉬를 포함한 모델을 얻을 수 있으며, 이들 간의 곡면 표현의 사실성에 거의 차이가 없다. 차량 모

델에서 대량의 메쉬를 생성하는 부분은 주로 작은 곡면 부품들로 구성되어 시각 거리를 고려할 경우 실제적인 표현 능력의 차이는 크지 않다.

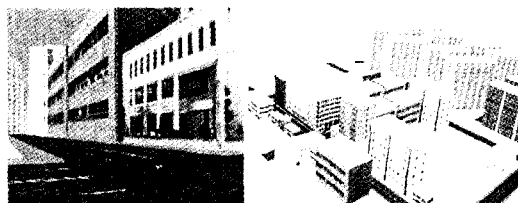


Fig. 9 Graphic model of virtual environment

Table 1 Statistics of environment and vehicle models

Model	Env.	Vehicle	
		iv (minimized)	iv (simplified)
File format	3ds	iv (minimized)	iv (simplified)
Frame rate [frames/sec]	30.0	15	30
Number of triangles	5,460	23,967	9,852
Number of nodes	46	91	60

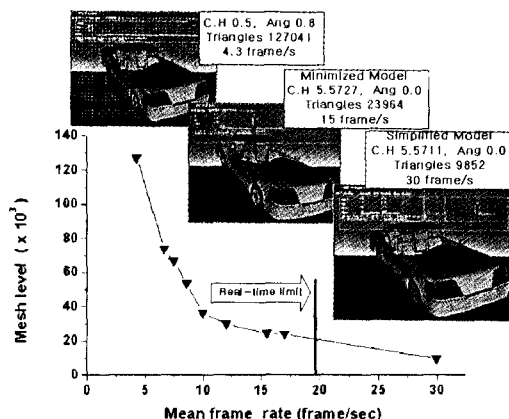


Fig. 10 Real-time mesh level test

본 연구에서는 차량의 형상적, 물리적 특성을 가상 프로토타이핑을 통해 얻는 것으로 가정하였

다. 가상 프로토타이핑을 통해 조립된 차량모델은 시각적으로 나타나지 않는 부분의 형상모델을 포함하고 있다. 가상 원형 및 설계 검증 등을 위한 모델은 차량 객체 전체를 직접 사용하여, 15 프레임/초 이상의 화면 갱신이 가능한 완전조립 모델인 23,964개의 삼각형 메쉬로 구성된 인벤토리 포맷의 차량 모델을 생성하였다. 주행 시물레이션에 사용되는 모델은 운전자 시야를 벗어나는 객체의 형상은 제외시켜 CULL과정에서 발생하는 계산량을 줄일 수 있다. 그러나 가상 원형 생성과정으로부터 전달되는 질량과 같은 객체의 물리적 정보는 소유관계에 있는 차체에 포함된다.

환경모델의 경우 100개 이상의 텍스처 이미지가 사용되었다. 이미지의 해상도를 높여 텍스처 전용 메모리의 용량을 초과할 경우는 화면 갱신율의 저하를 유발한다. 평균 시각 거리에 따른 이미지의 해상도를 조절하여 메모리 처리 용량을 벗어나지 않도록 조절하였다. 3차원 그래픽 파일을 불러들일 경우 각 부분을 하나의 노드로 표시하는데 환경모델의 경우는 46개의 노드로 구성되어 있으며 차량모델은 91개의 노드로 구성되었다.

4.2 병렬 구조 인터페이스

각 객체의 그래픽 처리와 더불어 실시간성과 몰입감에 직접적인 영향을 주는 인자로 가상현실 장비들과의 인터페이스가 있다. 패스트랙은 워크스테이션과 RS-232C 시리얼 포트를 통해 115,200 보드(baud)의 통신 속도로 데이터를 전송하도록 하고 시물레이션 프로그램의 갱신 과정내에 패스트랙의 데이터를 읽어오는 부분을 첨가하여 시점변화를 위한 데이터를 읽어오도록 구성하였다. 115,200 보드는 본 연구에 사용된 워크스테이션인 옥테인과 패스트랙에서 지원하는 최고의 속도이다. 패스트랙은 120 Hz로 동작하지만 4개의 센서가 직렬적으로 동작하므로 실질적으로 각각 30 Hz이하의 데이터 갱신율을 나타낸다. 단일 프로세스로 구성된 소프트웨어의 구조에서는 30.0 프레임/초의 화면 갱신율을 충족시키는 가상환경에 대해 패스트랙을 사용한 경우에

는 10 프레임/초 이하의 화면 갱신율을 나타내었다. 패스트랙 또는 데이터 클러브와 같은 장치들로부터 데이터를 읽어 들이는데 소요되는 대기시간 등 랜더링과 같은 계산적인 부하와 무관한 시간 지연이 발생하여 갱신율의 떨어진다. 이러한 지연 요소는 이미지의 깜박임과 같은 현상을 유발하여 현실감을 감소시키고, 참여자에게 부작용을 일으킬 수 있다.

이러한 가상현실 장비와의 통신 및 운동감 시스템과의 통신 등의 과정이 그래픽 갱신 속도에 주는 영향을 최소화 하기 위해서는 각 프로그램 모듈의 병렬 처리가 요구된다. 다중 쓰레드(multithread)¹⁷⁾를 사용하여 랜더링 루프와 다른 기능적 루프들을 분리시키는 구조는 이러한 단일루프에서 나타나는 시간지연 효과를 개선할 수 있다. 하나의 중앙 처리 장치를 가진 워크스테이션에서 시분할을 통해 여러 개의 LWP(light weight process)들을 구동하여 다중 처리와 유사한 기능적 동작을 실현할 수 있다. 그래픽 루프와 시리얼 통신 프로그램의 병렬 처리 시험 결과로부터 가상현실 장비와의 인터페이스 및 지연을 발생시키는 기능 함수들의 동작을 분리하여 화면 갱신 속도를 향상시킬 수 있는 구조를 적용하였다. Fig. 11은 다중 쓰레딩을 이용한 병렬 구조를 적용한 결과를 보여준다. 환경모델과 차량을 포

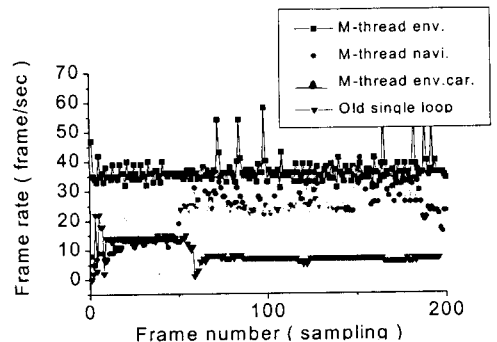


Fig. 11 Frame rate performance (SGI Octane, SI)

함한 환경모델, 그리고 가상 네비게이션을 수행

하는 경우에 대해 단일 루프 구조의 시뮬레이션에서의 화면 갱신율을 비교하였다. 결과로부터 기존의 단일 루프로 구성된 시뮬레이션 과정에서 발생하는 지연요소가 제거되어 갱신율의 향상효과를 얻을 수 있음을 알 수 있다.

5. 결론

본 연구에서는 자동차 시뮬레이터에 사용될 차량과 가상환경 모델을 개발하였다. 가상현실 기술을 차량 시뮬레이터에 적용하여, 6자유도 공간 센서인 패스트랙을 통해 운전자의 시점을 측정하여 시점의 변화를 가상환경에 반영할 수 있음을 확인하였다. 운전자에게 현실감과 몰입감을 주기 위해 헤드 마운티드 디스플레이를 사용하였으며, 운전자 머리의 회전과 손의 위치 및 손가락 자세 검출을 통해, 가상환경 내에서의 상호작용 및 입력을 가능한 가상현실 인터페이스를 구성하였다. 가상 차량과 환경 모델을 생성하기 위해 객체지향 모델링 기법중의 하나인 OOAD 표기법을 이용하였고, 이러한 객체 모델을 기반으로 주행 환경에 이용할 수 있는 가상 그래픽 모델을 생성하였다. 이는 차량 모델과 환경의 변화에 대한 정보가 변할 경우 효율적으로 수용하는 유연성을 가지며, 각 객체에 추가적인 속성과 동작을 부여할 수 있는 적응성을 가지게 된다. 가상 그래픽 모델은 객체 모델을 기초로 그래픽 라이브러리의 노드 구조로 구성한 후, 일반적인 모델링 프로그램과 프로 엔지니어를 이용하여 생성하였다. 생성된 환경 및 차량 모델을 그래픽 라이브러리를 이용한 시각제시 프로그램으로 가시화 하여, 환경 모델의 경우 실시간 시뮬레이션을 만족시키는 수준의 화면 갱신율을 얻을 수 있었다. 차량모델의 메쉬 평가를 통해 설계검증을 위한 모델, 실시간 주행 시뮬레이션을 위한 모델 등 목적에 맞는 수준으로 메쉬 정보를 다층화 할 필요성을 확인하였다. 가상현실 인터페이스의 구성에 있어서 병렬 구조를 적용하여 기존의 시뮬레이션에서 문제되는 시간 지연 요소들을 제거시켜 화면 갱신을

을 개선 시켰다. 향후 운동감 제시 시스템과의 연결을 통해 실시간 주행과 시나리오 주행 시뮬레이션을 평가하는 연구가 필요하다.

후 기

본 연구는 98년도 과학재단 특정연구 (97-0200-10-01-5)의 지원으로 수행되었음.

참 고 문 헌

- 1) J. Drosdol and F. Panik, "The Daimler-Benz Driving Simulator: A Tool for Vehicle Development," SAE paper 850334, pp. 1-17, 1985.
- 2) W. Käding and F. Hoffmeyer, "The Advanced Daimler-Benz Driving Simulator," SAE paper 950175, 1995.
- 3) J. Kuhl, D. Evans, Y. Papis, R. Romano, and G. S. Watson, "The Iowa Driving Simulator: An Immersive Research Environment," IEEE Computer Magazine, 1995.
- 4) E. J. Haug, K. K. Choi, J. G. Kuhl, and J. D. Wargo, "Virtual Prototyping Simulation for Design of Mechanical Systems," ASME Journal of Mechanical Design, Vol. 117, pp. 63-70, 1995.
- 5) 박민규, 이민철, 손권, 한명철, 유완석, 이장명, "차량 주행 감각 재현을 위한 운전 시뮬레이터 개발에 관한 연구 (I)," 한국정밀공학회 논문집 16권 6호, pp. 90-99, 1999.
- 6) 손권, 최경현, 유창훈, "건설 차량 실시간 그래픽 주행 시뮬레이터," 한국정밀공학회 논문집 16권 7호, pp. 109-117, 1999.
- 7) W. S. Lee, J. H. Kim, and J. H. Cho,

- "Development of a Driving Simulator,"
 Proceeding of the 9th International
 Pacific Conference on Automotive
 Engineering, Vol. 2, pp. 13-18, 1997.
- 8) 이승준, 김영삼, 조권주, 김일한, 최동찬,
 조준희, 이운성, 김정하, "실시간 고정
 베이스 차량 시뮬레이터 개발," 한국자
 동제어공학회 1998년 추계학술 대회논
 문집, pp. 2027-2030, 1998.
 - 9) J. Cremer, J. Kearney, and Y. Papelis,
 "Driving Simulation Challenges for VR
 Technology," IEEE Computer Graphics
 and Applications, pp. 16-20, September
 1996.
 - 10) Silicon Graphics Computer Systems, IRIS
 Performer™ Programmer's Guide, 1997.
 - 11) Silicon Graphics Computer Systems, IRIS
 Performer™ Class Reference Guide for
 C++ Programmers, 1997.
 - 12) Polhemus, 3Space Fastrak User's Manual,
 November, 1993.
 - 13) Fifth Dimension Technologies, '95 The
 Data Glove for the Fifth Dimension
 User's Manual, January, 1996.
 - 14) Virtual Research Systems Inc., V6 User's
 Guide, July, 1997
 - 15) Silicon Graphics Computer Systems,
 OCTANE™ Channel Option Installation
 Guide, 1997.
 - 16) G. Booch, *Object-Oriented Analysis and
 Design with Application*, Addison-Wesley,
 1994.
 - 17) S. Kleiman, D. Shah, and B. Smaalders,
Programming with Threads, Prentice-Hall,
 1996.