

# 가상현실을 이용한 실시간 차량 그래픽 주행 시뮬레이터

장재원\*, 손권\*\*, 최경현\*\*\*, 송남용\*

## A Real-Time Graphic Driving Simulator Using Virtual Reality Technique

Cheon Won Chang\*, Kwon Son\*\*, Kyung Hyun Choi\*\*\* and Nam Yong Song\*

### ABSTRACT

Driving simulators provide engineers with a power tool in the development and modification stages of vehicle models. One of the most important factors to realistic simulations is the fidelity obtained by a motion bed and a real-time visual image generation algorithm. Virtual reality technology has been widely used to enhance the fidelity of vehicle simulators. This paper develops the virtual environment for such visual system as head-mounted display for a vehicle driving simulator. Virtual vehicle and environment models are constructed using the object-oriented analysis and design approach. Based on the object model, a three-dimensional graphic model is completed with CAD tools such as Rhino and Pro/ENGINEER. For real-time image generation, the optimized IRIS Performer 3D graphics library is embedded with the multi-thread methodology. The developed software for a virtual driving simulator offers an effective interface to virtual reality devices.

**Key Words** : Vehicle Simulator(차량 시뮬레이터), Virtual Reality(가상 현실), Virtual Environment(가상 환경), Visual System(가시화 시스템)

### 1. 서론

최근 차량 개발 기간의 단축을 위해 동시공학 개념을 도입하여 기존의 수직적 개발 단계 요소들로부터 발생하는 시간지연을 제거하려는 많은 연구가 진행중이다. 셀(cell)이라 불리는 각 부분 모델 개발 주체들로부터 생성된 데이터를 네트워크를 통해 전송 받아 가상 원형(virtual prototype)을 생성하고, 이를 시험 및 평가함으로써 자동차 모델 개발에 유연성을 부여하고 생산 시간을 단축할 수 있다<sup>(1)</sup>.

자동차 주행 시뮬레이터는 검증된 자동차 모델 원형을 통하여 실제 모델 제작 이전에 주행 시뮬레이션이 수행할 수 있는 도구이다. 여러 가지 기계적 특성을 실험실 단위에서 검증함으로써 시작차의 제작 및 시험에 소요되는 많은 비용과 시간을 줄이는데 활용될 수 있다. 또한, 극한상황 시험, 사용자 훈련, 홍보뿐만 아니라 감성공학적 측면에서의 소비자의 취향과 요구를 분석하여 설계에 반영하는 등 여러 목적으로의 사용이 증대되고 있다. 벤츠 자동차<sup>(2,3)</sup>와 아이오와 대학<sup>(4,5)</sup> 등 선진연구기관을 중심으로 차량 시뮬레이터가 개발되어 자동차의 설계 및 시험에 적극적으로 활용되고 있다. 국내에

\* 부산대 대학원 기계설계공학과

\*\* 부산대 기계공학부

\*\*\* 제주대 생산에너지기계공학부

서도 대학, 기업체, 연구소 등에서 항공기 및 중장비 시뮬레이터<sup>(6,7)</sup>를 개발하였으며, 최근에는 자동차 시뮬레이터<sup>(8,9)</sup>의 개발에 대한 연구가 진행되고 있다.

일반적으로 자동차 주행 시뮬레이터는 시청각, 운동감, 조종감 등의 모의 제시를 통해 자동차의 운전상황을 피시험자에게 제공한다. 그러나 근사적 모의 한계와 더불어 현실감 있는 제시를 위해서는 많은 비용이 요구되며, 피시험자의 심리상황에 따른 모험적 행위 등 다양한 위험 요소를 내포하고 있다. 가상현실을 이용한 주행 시뮬레이터는 고가의 제시 장치를 사용하지 않고도 몰입감을 부여하여 현실감을 증대시킬 수 있으며 위험 요소를 사전에 배제시킬 수 있는 장점을 가진다. 또한, 가상현실 주행을 통해 생성된 주행정보는 동역학 해석을 거쳐 주행 시나리오로 활용될 수 있다. 주행 시나리오는 특수한 상황을 재현하거나 반복재현을 통하여 한정된 요인에 관련된 실험적 연구에 활용될 수 있다. 오프라인으로 해석된 시나리오 데이터는 위시아웃 필터와 역기구학 해석 등을 거쳐 시청각 제시시스템과 6자유도 운동 재현기를 통해 물리적인 주행 상황을 재현하는데 사용될 수 있다.

본 연구에서는 자동차 시뮬레이터와 사용자간의 가상현실 인터페이스 환경을 이용한 기구학적 주행을 통해 주행 시나리오 생성이 가능한 실시간 그래픽 소프트웨어(virtual reality driving simulator, VDS)의 개발에 대하여 기술한다. VDS는 독립적으로 그래픽 기반의 기구학적 주행 시뮬레이션을 수행할 수 있으며, 통합 환경에서는 중장비 시뮬레이터 개발을 통해 구성된 운동 재현기와 현재 개발이 진행중인 신형 제어기 및 동역학 해석기와 연동되어 동작하게 된다.

## 2. 그래픽 시뮬레이터 시스템

자동차 시뮬레이터는 Fig. 1과 같이 크게 운동 재현기, 제어기, 동역학 해석기, 3차원 그래픽스 및 운영 시스템 부분으로 나누어진다. 운동 재현기는 실제 차량 운동에 의해 생성되는 가속도를 제공하기 위해 사용되며, 한정된 범위 내에서 유압시스템에 의한 3방향의 병진 운동과 3축 중심의 회전운동이 가능하도록 되어있다. 유효 하중이 크고 안정성이 뛰어난 스텐더트 플랫폼 구조의 운동 재현기 상판에는 실제 차량과 조작기구들이 탑재되어 있

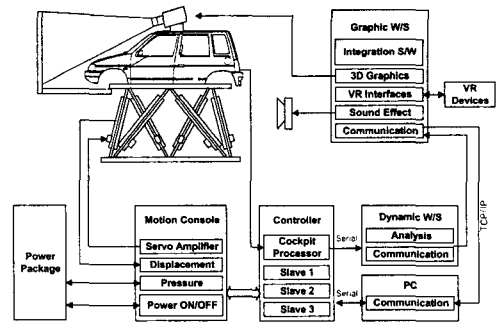


Fig. 1 Vehicle driving simulator

다. 전용 제어기는 운동 재현기의 자세 제어와 워크스테이션과의 통신을 수행한다. 조작장치의 조작신호는 센서를 이용해 검출하고 동역학 해석기는 입력값에 대한 차량의 상태를 해석하여 결과를 전달한다. 그래픽 주행 소프트웨어는 워크스테이션에 탑재되어 가상현실 인터페이스를 통해 고속 영상 처리를 수행하고, 동역학 해석기 및 운동 재현기를 위한 제어기와 실시간으로 정보를 전달하며 시스템을 관장한다.

그래픽 시뮬레이터는 시각적 현실감 제공과 함께 실시간 렌더링 및 운영, 운동 재현기 및 해석기와와의 자료 전달, 고정 시뮬레이터로서의 기구학적 주행, 가상현실 장치 입출력, 후처리 및 평가 등 다양한 기능이 요구된다. 따라서 그래픽 시뮬레이터가 현실감 제공에 가장 큰 영향을 주는 요소이다.

## 3. 그래픽 모델

차량 주행 시뮬레이터에 사용되는 그래픽 모델은 설계 자료로부터 생성된 차량 모델, 주행환경 모델, 그리고 가상 공간에서 운전자를 대행하는 대행체(avatar)로 구성된다. 대행체는 가상공간에서 운전자의 손을 표현하고 시점의 위치를 결정하는 인체모델의 형상 치수로 결정된다. 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 OOAD<sup>(10)</sup> 기법을 이용하여 객체를 모델링하고 객체들 사이의 관계를 정의하였다. 그리고 모델링된 객체는 그래픽 라이브러리와 3차원 모델링 도구를

이용하여 그래픽 모델을 생성하여 가상환경에 적용하였다.

### 3.1 설계 모의와 차량 모델

자동차 개발 단계에서 설계과정은 부품 설계로부터 통합, 시험, 평가 등에 이르는 과정을 거친다. 이러한 과정에서 가상 원형은 부품단위의 설계 정보로부터 조립 및 동작을 모의 실험하는 설계검증에 활발히 도입되고 있다. 이러한 자동차의 동시공학 설계과정은 Fig. 2와 같이 3개의 계층으로 나누어 설명할 수 있다. 자동차 모델 개발 단계에서 개별 설계 주체들로부터 생성된 데이터는 보다 상층에 존재하는 가상 원형 생성을 위한 dVISE와 같은 가상현실 통합 시스템에 전달되어 시험된다<sup>(11)</sup>. 자동차 시뮬레이터 시스템은 이러한 가상 원형과 설계 객체로부터 전송된 물리적 정보를 직접 또는 변환을 거쳐 시뮬레이션 정보로 사용할 수 있다. 실시간 시뮬레이터의 데이터 베이스에는 차량객체와 환경객체에 대한 형상정보를 담고 있는 메쉬 정보와 그 속성, 텍스처 좌표, 그리고 이미지 처리과정을 거친 텍스처 소스가 포함된다. 가상 원형으로부터 얻어지는 물리정보와 기능정보는 객체의 행위를 규정하기 위해 시뮬레이터로 전달되어야 한다. 데이터 베이스가 시뮬레이션 계층에 전달되면 평가목적에 따라 정의된 프로세스를 수행한다. 주행 시뮬레이션이 수행되는 동안 가상현실 주행 시뮬레이터는 평가와 후처리를 위해 실시간으로 차량 위치, 객체의 상태 변화, 차량의 주행 등의 정보를 저장한다. 이러한 데이터는 대상 인자에 대한 영향을 고려하여 설계 인자를 수정하는데 사용된다.

본 연구에서는 차량 및 주행환경에 대해 객체와 객체 사이의 관계, 각 객체의 속성과 기능을 정의하여 객체 모델을 생성하였다. 차량 및 환경 정보는 정의된 객체들의 집합인 클래스로 분류하고, 각 클래스 사이의 관계를 정의하여 실세계의 속성과 기능을 반영한 모델을 생성하였다.

설계 원형의 정보를 폴리곤을 기반으로 한 실시간 렌더링에 사용하기 위해서는 메쉬 포맷으로의 변환 필요하다. 설계단계를 모의하고 가상환경에서 사용될 형상 정보를 생성하기 위해 본 연구에서는 통합 설계 소프트웨어인 프로엔지니어(ProENGINEER), 님스(NURBS)기반의 CAD모델러인 라이노3D(Rhino3D), 그리고 3D 스튜디오 맥스(3D Studio MAX, R2.0)를 사용하였다. 설계 정보를 기초로 객체 모델에 따라 부품들을 형상화하여 조립하고, 각 부품들에 대해 재질과 텍스처 맵핑 좌표를 부여하였다. 부품 객체와 차량에 대한 기능적인 동작 함수들을 정의하고, 주행 시뮬레이션에서 입력될 수 있는 인자들과의 상호작용을 정의하였다. VDS의 데이터베이스에는 정교한 프라모델의 치수정보로 구성된 중형자동차와 운동 재현기에 탑재되는 경승용차의 객체 모델이 포함되어 있다.

### 3.2 주행 환경 그래픽 모델

시뮬레이션이 수행될 모의 환경을 정의하고 이를 사실적으로 생성 및 제공하기 위한 환경 모델 데이터 베이스가 필요하다. 그러나 형상 모델링에 의존한 주행 환경 객체는 과도한 폴리곤과 객체의 재질 정보를 필요로 하며, 제작시 많은 시간과 노력이 요구된다. 보다 적은 폴리곤을 사용하여 대규모 가상세계를 사실적으로 표현하기 위해 디지털 카메라로부터 얻어진 실세계의 영상에서 추출한 텍스처 맵핑 이미지를 사용한다. 원근 효과, 렌즈 왜곡, 장애물 등은 이미지 변환 및 필터링 과정을 통해 제거된다. 시점으로부터의 거리에 따라 맵핑 이미지의 해상도를 다중화 하여 사용하는 밍맵(mipmap)의 지원을 위해 2<sup>n</sup> 픽셀 단위로 이미지를 생성하였다. 실세계 대상의 주행 모델은 실차 시험과 시뮬레이션 결과와의 비교, 수요자의 선호도 등과 관련한 정보 생성에 유용하다. 이를 위해 선정된 주행 환경에 대한 건축 도면 정보에 따라 특징적 형상을 추출하여 건물을 모델링 하고, 추출된 텍스처 맵을 지정면에 맵핑하여 텍스처 좌표 정보

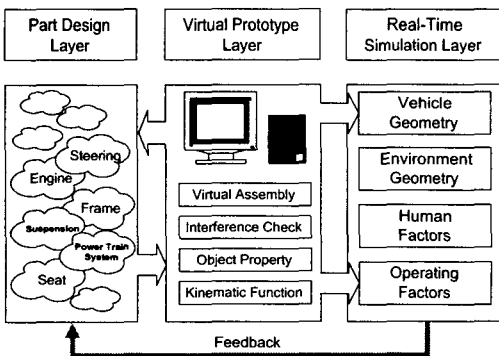


Fig. 2 Data transferring layer

를 생성하였다. 또한, 항측도를 참조하여 지형 및 환경 데이터 베이스의 위치를 결정하여 주행 환경 모델을 사실적으로 생성하였다. Fig. 3은 100개의 텍스처, 13,000개의 메쉬, 197개의 객체 모델로 구성된 주행환경 모델을 보여준다.

유한한 주행 환경의 경계를 처리하기 위해 환경 주변의 디지털 이미지를 15° 간격으로 얻어 이를 연속 이미지의 파노라마 맵을 구성하였다. 경계는 실제계에서의 크기 비에 따라 평면으로 처리된 환경 맵을 실린더 형상의 경계면에 대한 맵핑으로 처리하였다. 지형 모델은 항측도에서 얻어진 등고선 형상으로부터 모델링되었다.

#### 4. 가상현실 인터페이스

가상현실 인터페이스는 일반적인 입출력 환경을 제공할 뿐만 아니라 사용자에게 현실감 증대시키고 몰입감을 생성시키기 위하여 사용된다. 실제계에서의 조작 행위와 동일한 방법으로 시스템과의 상호작용도 가능하게 해준다. 그러나 인체의 운동은 자유도가 높고 비선형 요소를 다수 포함하고 있으며, 가상현실 시스템의 실시간 유지를 위해서는 이러한 요소를 최소화해야 한다. 그리고 일반적으로 제품화된 가상현실 장비들이 검출할 수 있는 자유도는 인간의 자유도에 비해 작으므로 근사적 표현이 불가피하다. 하드웨어 인터페이스는 물리적으로 2개의 RS-232 직렬 통신 포트, 이더넷 포트, 그리고 3가지 비디오 신호 출력 포트를 통해 이루어진다. VDS 소프트웨어는 각 장치 모듈을 독립

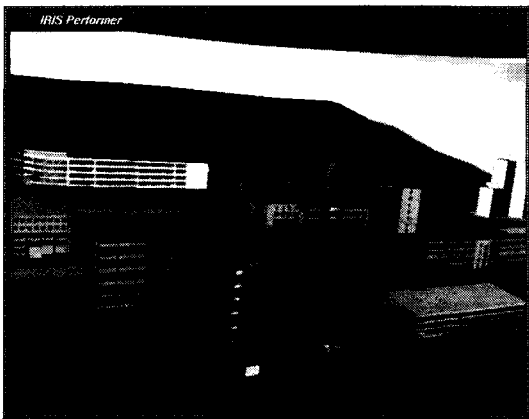


Fig. 3 Environment model

적으로 구동하고 공유 영역을 통해 비동기적으로 정보를 전달하는 구조를 가진다.

#### 4.1 하드웨어 인터페이스의 구성

가상현실 장비를 포함한 그래픽 시뮬레이터의 하드웨어 인터페이스의 구성은 Fig. 4와 같다. 자동차 시뮬레이터를 위한 동역학 해석기와 운동 재현기는 허브(HUB)를 통해 연결되며 TCP/IP 프로토콜을 이용하여 정보를 전달한다. 본 연구에서 그래픽 시뮬레이터 하드웨어는 가상현실 장비와 출력 장치를 포함한 워크스테이션으로 구성된다.

패스트랙(Fastrak)<sup>(12)</sup>은 공간상의 위치와 방위를 측정하는 6자유도 센서로 전자기파를 방출하는 트랜스미터로부터의 상대적인 거리 및 방위를 측정한다. 탑승자의 이동 범위 이내에서는 일반적인 실험실 환경에서 발생하는 노이즈에 의한 오차가 가시적 범위를 넘지 않는다.

데이터글러브로 사용된 5th-Glove<sup>(13)</sup>는 광섬유를 사용하여 각 손가락의 굽힘량을 검출한다. 5자유도 센서로 손가락의 각 마디의 움직임을 얻을 수 없으므로 비례적인 기준값에 대한 보정이 요구된다. 본 연구에서는 초기상태에서 최대 굽힘과 완전 이완으로부터 얻어진 값을 기준으로 각 손가락에 대해 선형적인 관절 각도 변화를 부여하여 가상 환경에서 운전자의 손동작을 표현한다.

HMD는 트랙커(tracker)를 부착할 경우 다른 디

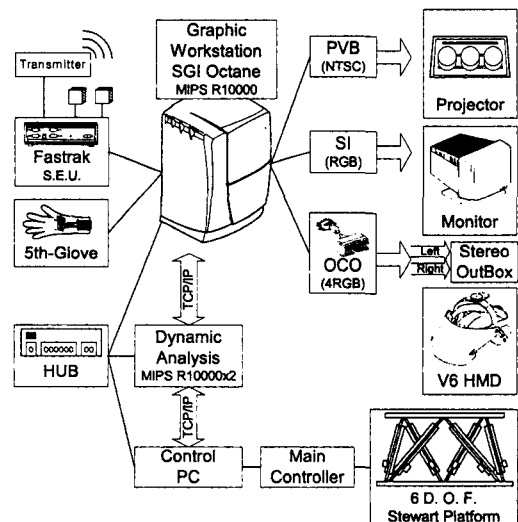


Fig. 4 Hardware interface in driving simulator

스플래이 장치들과 달리 전방위의 표현이 가능하며, 사용자에게 몰입감을 제공해 주는 장점이 있다. 본 연구에서는 Virtual Research System사의 V6 HMD<sup>(14)</sup>를 사용하였다. V6는 2개의 1.3 인치 LCD를 장착한 반 헬멧형상의 시각정보 출력 장비로, 스크린과 프로젝터를 사용하는 시스템이 시점변화에 대응할 수 없는 단점을 보완할 수 있다.

워크스테이션이 가지는 그래픽 출력 장치는 기본적인 그래픽 출력 이외에 NTSC신호를 얻을 수 있는 PVB(personal video board)<sup>(15)</sup>와 여러 RGB신호를 출력할 수 있는 다중 채널 옵션(multi-channel option)<sup>(16)</sup>을 가진다.

가상현실 입출력 장비들은 워크스테이션과 RS-232C 시리얼 통신을 통해 데이터를 주고 받는다. 각 장비에서 측정된 데이터는 실세계와 가상환경 등에서 정의되는 좌표계와 일치하지 않으므로, 좌표축 변환 및 스케일 조절을 통해 측정값을 가상 공간에서의 좌표축과 일치시켜 사용된다. 구성된 가상현실 인터페이스는 일반적인 실험실 환경에서 전자기 트랜스미터와의 거리가 1 m 이내일 때 운전자 시점의 변화에 대응시킨 결과, 패스트랙은  $\pm 60^\circ$ 의 전방시야에 대해 안구 운동을 고려하지 않을 경우  $0.7^\circ$  이내의 오차로 시점 변화를 반영할 수 있었다.

#### 4.2 병렬처리 구조

단일 루프로 구동되는 시뮬레이터 소프트웨어에서는 주변 장치와의 직렬 통신과 같은 기능적 모듈이 시뮬레이션 루프의 갱신에서 지연요소로 작용하는 단점이 있다. 동일한 하드웨어 상황에서 단일 루프 시스템은 패스트랙(Fastrak) 센서를 포함하여 10 frames/sec이하의 갱신율을 나타내었다. Fig. 5는 다중 쓰레딩(multi-threading)을 이용한 VDS의 병렬 프로세스 구동 구조를 나타낸다. 다수의 LWP(light weight process) 생성을 통해 각 기능 모듈을 독립적으로 구동시키고, 공유 자원 설정을 통해 데이터의 갱신을 실현함으로써 렌더링 루프에서의 지연요소를 최소화할 수 있다.

이러한 방식의 구조에서는 공유자원에 대한 프로세스들의 동시 접근 문제, 시스템의 동기화 문제, 데이터 손실 등의 문제가 발생할 수 있다<sup>(17)</sup>. 시스템에서는 동시 접근 문제를 해결하기 위해 상호 배제를 통해 자원 접근을 제한하였으며, 2차적인 접근을 통해 프로세스의 잠금 시간을 줄였다. 상호

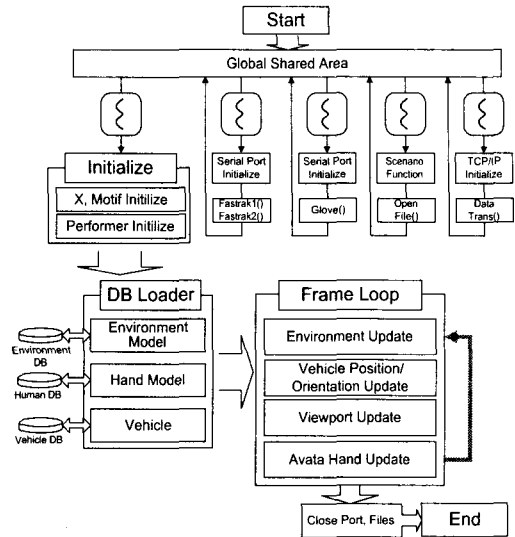


Fig. 5 VDS operation structure

배제에 의한 데이터의 손실은 가시화 측면에서 인지범위 이하의 영향을 보였다. 동일한 환경과 차량 모델에 대하여 다중 쓰레드를 사용한 경우에서 환경 렌더링, 네비게이션, 환경과 차량 렌더링을 수행한 결과, 모든 경우에 대해 20 frames/sec 이상의 갱신율을 유지할 수 있었다. 단일 프로세스로 구성된 기존의 시뮬레이션 루프에서 발생하는 약 50 ms의 시간지연 요소를 배제시킬 수 있다.

#### 4.3 계층적 좌표 구조

VDS시스템에서의 객체 위치는 동적 좌표계(dynamic coordinate system, DCS)와 정적 좌표계(static coordinate system, SCS)로 정의되며 계층적인 구조를 가지고 있다. 동적 좌표계는 프로그램 갱신에 따라 각 프레임마다 그 위치를 계산하여 장면에서 그 객체의 움직임을 표현한다. 정적 좌표계는 환경 모델과 같이 가상환경내의 절대 좌표에 대해 그 위치가 변하지 않는 객체의 초기 위치 지정에 사용되며, 이는 시스템의 계산량을 감소시킬 수 있다. Fig. 6은 구성된 객체의 계층구조를 보여준다.

가상환경 참여자에 대해 부분적인 인체 모델을 도입하여 가상환경내의 대행체(avatar)를 정의하고, 센서를 통해 머리와 손의 움직임을 검출하여 가상환경내의 행위를 부여하였다. HMD 착용자는 현실 세계 정보와 차단되며 가상세계의 입력을 위해 데

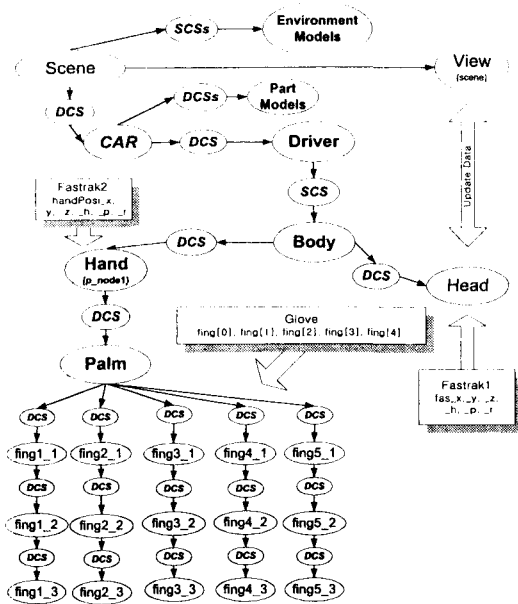


Fig. 6 Hierarchical coordinate of object structure

이터 글러브를 사용한다. 손의 형상 모델에 대해 17개의 동적 좌표를 부여하고, 사용자에게 따라 손가락을 편 상태와 구부린 상태를 학습시켜 손의 동작을 입력한다. 시점 결과와 손 위치와 방향의 검출을 위해 2개의 6자유도 공간센서를 각각 HMD와 데이터 글러브에 부착하였다. 실제세계와의 일치성을 위해 초기 설정 단계에서 기준 위치를 제공하여 사용자의 감각적 일치감을 반영할 수 있게 하였다. 기준 위치는 인체모델 생성 패키지인 ADAMS/Android<sup>(18)</sup>와 GEBOD<sup>(19)</sup>에서 생성한 표준 인체로부터 얻을 수 있다. 조향 장치와 같은 조작기와의 상호작용을 위해 특징적인 형상 경계를 부여하고, 조작기와의 거리에 해당하는 반경검사의 결과에 따라 이벤트를 정의한다.

#### 4.4 그래픽 가시화 시스템

다양한 영상제시 기술의 개발에도 불구하고 현재까지 실제세계의 사실성을 완전하게 대체할 수 있는 표시장치기술은 존재하지 않는다. 고해상도의 현실감 있는 영상을 제공하기 위해서는 기본적으로 컴퓨터에서 생성된 원 이미지의 해상도가 높아야 한다. 그러나 이미 계산된 이미지의 출력은 표시장치의 해상도, 시야, 스테레오 이미지의 구현 방식 등 실질적인 표현 방식과 표시장치의 성능에 따라

현실감의 차이를 유발한다.

VDS의 가시화 시스템은 Table 1과 같이 크게 3가지의 출력 장치로 구성된다. 워크스테이션이 가지는 기본적인 그래픽 출력 신호와 PVB에서 얻어지는 NTSC 신호, 그리고 OCO(octane channel option)에 의한 4채널의 RGB신호를 출력할 수 있다. 본 연구에 사용된 워크스테이션의 기본 그래픽 출력은 1280×1024의 해상도와 72 Hz의 동작속도를 가진다.

3차원 렌더링의 경우 일반적인 출력장치들의 구성인 1:0.75 비의 영상에 대해 렌더링 이미지의 과도한 왜곡을 피하기 위해서 가로 세로 60°, 45°의 FOV(field of view)를 사용하는 것이 일반적이다. 그러나 인간의 수평, 수직 시야각은 180°, 135°에 이르며, 수평 시야에 대해 전체 시야를 포함할 수 있는 영상을 제시하기 위해서 3개 이상의 프로젝터와 실시간 렌더링 이미지의 출력이 요구된다.

좁은 수평 시야는 자동차 시뮬레이터에 있어서 심각한 속도감의 손실을 야기한다. 그러나 투사 이미지는 스테레오 이미지에 비해 깊이감이 떨어지며 전 방위 시각을 확보하기란 현실적으로 불가능하다. 반면, HMD의 넓은 시야를 확보하는 힘드나 모션 트랙커와 함께 사용하여 전 방향에 대한 시각을 제공할 수 있고, 가상 환경과 사용자가 실시간 상호작용하여 몰입감이 생성된다. 또한 스테레오 이미지를 제공하여 깊이감을 부여할 수 있다. VDS 소프트웨어는 다중 파이프, 다중 채널 실시간 렌더링을 구현하였으며, Fig. 7과 같이 3개의 렌더링 이미지를 생성하여 수평시야 180°의 영상을 제공할 수 있다. OCO는 프레임 버퍼에 적재된 출력 정보를 정해진 크기로 분할하여 각각 독립적인 다

Table 1 Graphic hardware specification

Graphic hardware	Output signal	Resolution	Channel	Function
SI	RGB	1280 × 1024	1	Default
PVB	NTSC /PAL	525 lines /625 lines	1	Real-time capture
OCO	RGB	4@ 640 × 480	4	Multi channel output

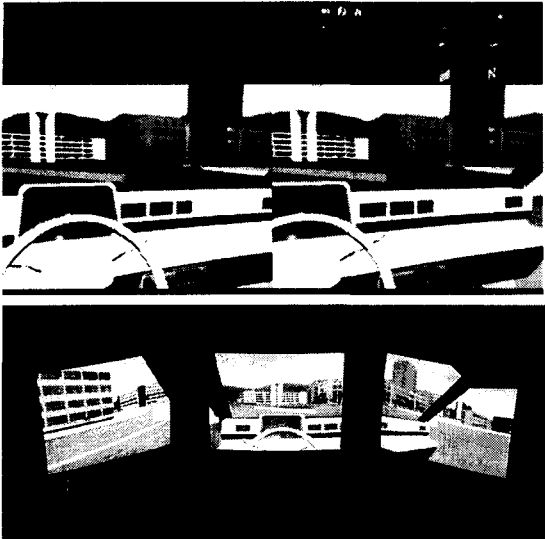


Fig. 7 Multi-channel visualization

수의 프레임 버퍼를 통해 내 보낸다. 그래픽라이브러리의 렌더링 영역에 있어서 하나의 그래픽 파이프는 다수의 채널을 가질 수 있으며 각 채널은 독립적인 시야절두체(view frustrum)로 구성된다. VSD는 Fig. 8과 같이 2채널의 스테레오 이미지와 180° 수평시야 제공을 위해 각각 2개의 채널과 1개의 채널을 가진 2개의 파이프 윈도우로 구성된 멀티 파이프, 멀티 채널 렌더링을 지원한다. 또한, VDS는 V6와 I-glass HMD로의 출력을 지원한다. 특히, V6 HMD는 좌, 우 독립입력이 가능하여 입체영상을 위한 스테레오 뷰(stereo view)를 구현 할 수 있다. 거리감 인지의 원인이 되는 양안 거리와 초점거리를 입력하여 2개의 채널에 각각의 이미지를 제공함으로써 입체감을 제공한다. VDS 소프트웨어는 고정 초점거리와 사용자의 양안 거리의 입력을 통해 계산된 양안 시차를 제공하여 스테레오 이미지를 생성 기능을 제공한다. 시점의 변화는 HMD에 부착된 센서로부터 검출된 사용자 머리의 방위 값을 통해 지속적으로 갱신된다.

### 5. 주행 시뮬레이션

VDS는 주행정보를 얻기 위해 동역학 해석기와 연결되어 동작한다. VDS는 이러한 정보의 전달과 실시간 처리를 위해 LWP로 동작하는 LAN통신 모듈

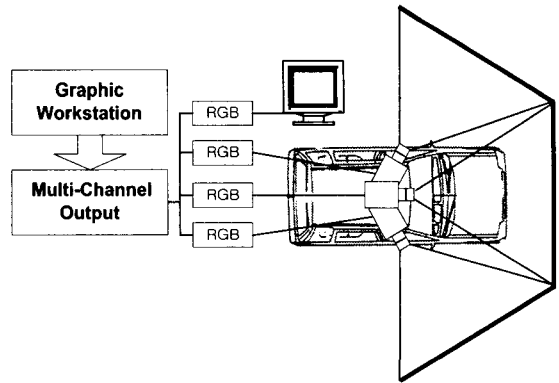


Fig. 8 Multi-projection for PNU driving simulator

을 포함하고 있다. 실시간 요구를 받지 않는 시나리오 데이터의 해석을 위해서는 범용의 동역학 해석 소프트웨어를 사용할 수 있다. 워크스테이션 환경에서 실시간 해석을 수행 할 수 있도록 개발중인 전용 소프트웨어와 연결되어 실시간 주행 시뮬레이션을 수행 할 수 있다. 또한, VDS는 자체에 계산적 부하가 현저히 낮은 기구학적 주행을 통해 실시간 그래픽 시뮬레이션을 수행 할 수 있는 기능을 포함하고 있다. 이러한 기구학적 주행의 결과는 오프라인으로 동역학 해석을 거쳐 VDS와 운동 재현기를 통해 재현될 수 있다.

### 5.1 기구학적 주행 처리

자동차의 기구학적 주행은 시간에 따른 운동 구속 조건을 부여하여 0자유도로 구현된다. 가속페달 입력에 대하여 자동차의 속도 구속 조건을 생성하기 위해 2차원의 차량 모델에 뉴턴 2법칙을 적용하여 주행 속도를 산출하였다. 차량의 가속력은 엔진으로부터 얻어진다. VDS는 엔진으로부터 생성되는 토크를 얻는 방법으로 2가지 모드를 가진다. 엔진 거동을 모사한 엔진식을 사용하는 모드와 엔진 시험을 통해 얻어진 데이터를 보간하여 사용하는 모드를 선택할 수 있다. 엔진 모델은 Cho<sup>(20)</sup>가 개발한 동력전달장치 모델을 사용하였다.

이전 루프로부터 생성된 차량의 속도와 가속 페달에 의한 스톱을 밸브 개폐량에 대하여 1차함수로 근사 표현되는 자동변속선도를 적용한다. 차축으로 전달되는 토크는 변속기, 토크컨버터, 그리고 최종 감속단을 거쳐 결정된다. 또한 구동계의 관성 효과와 마찰 등의 저항이 존재한다. 이들은 전달

효율로 대체하여 반영할 수 있다. 제동력은 생성되는 토크에 반대방향으로 작용하며 크기는 브레이크 페달 입력에 선형 비례하는 것으로 가정하였다.

일반적인 승용차에 적용되는 전륜 2WS 모델로부터 에크만 조향각과 자동차의 선회반경은 쉽게 얻을 수 있다. 구동력으로부터 얻어진 속도가 차량의 회전반경에 수직으로 작용한다고 가정하면, 주어진 속도에 대해 짧은 시간 후의 차량 위치와 방향은 순간 중심에 대한 기구학적 관계로부터 구해진다. 주행 상황에서 발생하는 모든 이벤트 신호와 차량의 거동은 실시간으로 저장되며, 주행 종료후 LAN을 통해 동역학 해석 시스템으로 전달되어 해석된다. 기구학적 주행의 결과는 주행 시나리오로 사용이 가능하다. 운전자의 입장에서 가상 주행한 상황에 대해 안전성과 재현성 평가를 거쳐 해석된 데이터는 운동 재현기를 포함한 주행 재현을 가능하게 한다. 운전자 모드는 VDS에서 제공하는 인터페이스를 사용하여 기구학적 가상 주행을 통해 주행 시나리오를 생성시킨다. 시나리오는 동역학 해석기로 전달되고 생성된 결과는 평가를 거쳐 운동 재현기와 VDS에 전달되어 운전 상황을 재연할 수 있다. 이때 피시험자는 가상차량 내의 승객 위치로 이동될 수 있으며 물과 피치 효과가 포함된 영상과 모의된 가속도를 제공받는다.

생성된 시나리오의 주행 경로 정보를 추종하기 위해 주행 경로에 대한 조향 입력의 제어를 위해 운전자 모델이 적용될 수 있다. 시나리오의 동역학 해석은 실시간 요구를 받지 않으므로, AutoDyn7<sup>(21)</sup>, ADAMS<sup>(22)</sup>와 같은 소프트웨어가 사용될 수 있다. AutoDyn7은 국내의 독자적인 차량동역학 해석용 소프트웨어로써, 운전자 모델을 적용하여 VDS가 생성한 시나리오를 추종한 결과를 제공할 수 있다. 기호연산기법을 적용하여 10초의 주행 시뮬레이션을 7초 이내에 해석할 수 있어, 실시간 시뮬레이터의 해석기로도 사용될 수 있다. ADAMS에서는 가상원형으로부터 전달된 물성치를 기초로 차량 모델을 생성하여 특정한 주행 상황을 해석하는데 사용될 수 있다<sup>(23)</sup>. Fig. 9는 카메라 뷰 모드에서 본 가상주행 상황을 보여주고 Fig. 10은 생성된 주행 시나리오 데이터를 나타낸다. VDS가 제공하는 인터페이스를 사용하여 기구학적 주행을 수행한 경로와 운전자 모델의 전방 주시 거리를 10 m로 하여 전방거리오차 제어방식을 통해 경로를 추종시킨 결과이다.

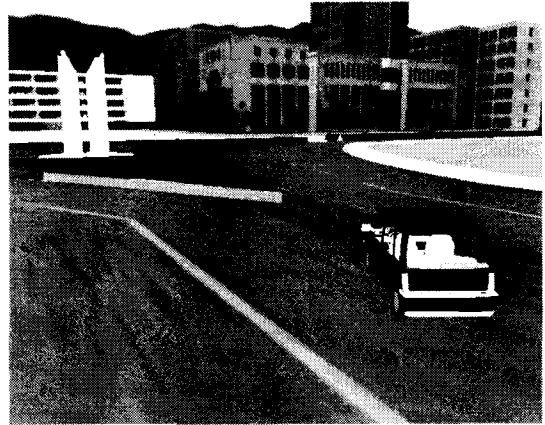


Fig. 9 Camera view

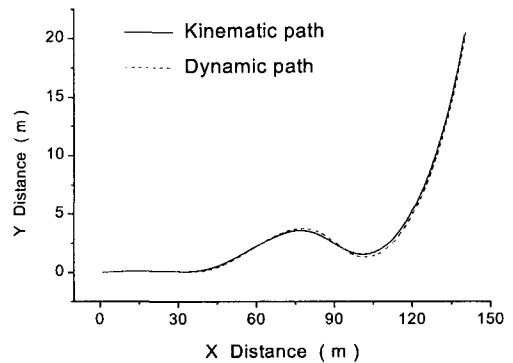


Fig. 10 Driving scenario

## 5.2 주행 시뮬레이션 소프트웨어

VDS 시스템의 개발에는 실리콘 그래픽스사의 워크스테이션(Octane, 195MHz)과 유닉스기반의 운영체제 IRIX 6.4에서 퍼포머(Performer)<sup>(24)</sup>, 모티프(Motif)<sup>(25)</sup> 라이브러리를 포함한 C/C++언어가 사용되었다. Fig. 11은 본 연구를 통해 개발된 VDS 소프트웨어의 동작 환경을 보여준다. 편리한 사용자 입력의 처리를 위해 X-WINDOW 환경에서의 GUI를 설계하고 그래픽 설정, 가상현실 장비 설정, 각종 환경 변수 입력 등의 기능을 구현하였다. 렌더링 영역과 각종 인터페이스를 분리하여 독립적으로 동작 할 수 있도록 하였다. 이러한 분산 구조는 불필요한 갱신을 최소화 시켜 시스템의 부하를 감소시킬 수 있는 장점을 가진다. 최근 리눅스(Linux)



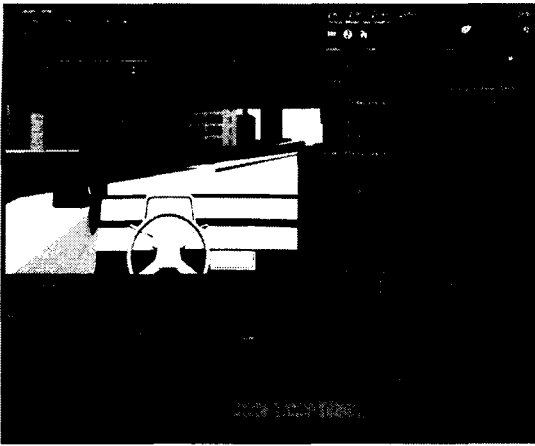


Fig. 11 VDS operation view

기반의 PC에서 사용할 수 있는 퍼포머 라이브러리가 무상배포 되고 있어 VDS는 다른 워크스테이션으로 뿐만 아니라 PC환경으로의 이식성이 크게 향상되었다.

## 6. 결론

본 연구에서는 가상현실 기술을 이용하여 그래픽 기반의 자동차 시뮬레이터 시스템을 개발하였다. 자동차 시뮬레이터를 위한 몰입형 가상현실 인터페이스를 구성하고, 보다 현실감 있는 대규모 가상 객체 데이터를 생성하였다. 또한 이들의 실시간 처리를 위해 다양한 알고리즘을 적용한 소프트웨어를 개발하였다. 다중 채널 장비와 다중 파이프 및 채널 프로그래밍을 통해 넓은 시야를 제공할 수 있으며, HMD를 통해 스테레오 이미지를 제공하여 운전자에게 현실감과 몰입감을 줄 수 있도록 하였다. 가상 차량과 가상환경 모델을 생성하기 위해 객체지향 방법론을 이용하여 객체 모델을 개발하고 형상화하였다. 그래픽 라이브러리의 계층적 노드 구조로 차량 및 환경 가상환경 내의 참여자 대행체의 상관관계를 모델링 하고, 이들간의 실시간 상호작용을 기능적으로 구현하였다. 편리한 사용자 인터페이스를 구축하였으며, 마우스와 키보드를 사용하여 메뉴 상에서 가상환경의 각종 선택사항을 편리하게 설정할 수 있도록 하였다. 가상 차량모델의 기구학적 주행 시뮬레이션을 구현하여 주행 시나리오를 생성하고 재현하였다.

차후 통합 시뮬레이터를 구성하여 자동차 개발 과정에서 설계 원형 시험, 특수 상황 재현, 인간 요소 평가 등의 목적으로 활용될 수 있을 것이다. 그리고 보다 현실적인 입력상황을 모의하기 위해서는 반력 생성을 포함한 물리적 조작기구의 개발이 필요하다.

## 후기

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

## 참고문헌

1. Michael, H., Virtual Prototyping of Vehicle Control Systems, International Symposium on Advanced University of Technology, pp. 1105-1124, 1996.
2. Drosdol, J. and Panik, F., The Daimler-Benz Driving Simulator: A Tool for Vehicle Development, SAE 850334, pp. 1-17, 1985.
3. Käding, W. and Hoffmeyer, F., The Advanced Daimler-Benz Driving Simulator, SAE 950175, 1995.
4. Kuhl, J., Evans, D., Papeis, Y., Romano, R., and Watson, G., The Iowa Driving Simulator: An Immersive Research Environment, IEEE Computer Magazine, pp. 35-41, 1995.
5. Haug, E. J., Choi, K. K., Kuhl, J. G., and Wargo, J. D., Virtual Prototyping Simulation for Design of Mechanical Systems, ASME Journal of Mechanical Design, Vol. 117, pp. 63-70, 1995.
6. 김승주, 김영철, KTV-1항공기 SIMULATOR 개발, 한국항공우주학회 1995년도 추계 학술강연회 초록집, pp. 364-367, 1995.
7. 손권, 최경현, 유창훈, 건설 차량 실시간 그래픽 주행 시뮬레이터, 한국정밀공학회 논문집 제16권 제7호, pp. 109-117, 1999.
8. Lee, W. S., Kim, J. H., and Cho, J. H., Development of a Driving Simulator, Proceeding of the 9th International Pacific Conference on Automotive Engineering, Vol. 2, pp. 13-18, 1997.

9. 김형내, 김석일, 20자유도 자동차모델을 이용한 가상 주행 시뮬레이터의 개발, 한국CAD/CAM 학회논문집, pp. 40-47, 1998.
10. Booch, G., Object-Oriented Analysis and Design with Application, Addison-Wesley, 1994.
11. Division Ltd. UK, dVISE User Guide, 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 Personal Video Installation Guide, 1997.
16. Silicon Graphics Computer Systems, Octane Channel Option Installation Guide, 1997.
17. Kleiman, S., Shah, D., and Smaalders, B., Programming with Threads, Prentice-Hall, 1996.
18. Mechanical Dynamics Inc., ADAMS/Android User's Guide, 1991.
19. Cheng, H., Obergefell, L., and Rizer, A., Generator of Body Data(GEBOD) Manual, Wright-Patterson Air Force Base, Ohio, 1994.
20. Cho, D., Nonlinear Control Methods for Automotive Powertrain Systems, Ph. D. Dissertation, MIT, 1987.
21. 김광석, 유완석, 김성수, 김상섭, 차량동역학 해석 프로그램 AutoDyn7의 개발(1) - 프로그램 구성 및 주요 알고리즘, 자동차공학회논문집, 제7권, 제3호, pp. 321-330, 1999.
22. 고정훈, 손권, 최경현, 승용차의 가상 프로토타이핑, 자동차공학회논문집, 제7권, 제5호, pp. 230~239, 1999.
23. Mechanical Dynamics Inc., ADAMS User's Manual, 1994.
24. Young, D. A., The X Window System Programming and Application with Xt, Prentice Hall, 1994.
25. Silicon Graphics Computer Systems, IRIS Performer Programmer's Guide, 1997.