

CAVE 상에서의 방대한 볼륨 데이터의 실시간 입체 영상 가시화

(Real-Time Stereoscopic Visualization of Very Large Volume Data on CAVE)

임 무 진 [†] 이 중 연 ^{**} 조 민 수 ^{***} 이 상 산 ^{****} 임 인 성 ^{*****}
(Moojin Im) (Joongyeon Lee) (Minsu Joh) (Sangsan Lee) (Insung Ihm)

요약 과학적 가시화의 한 분야인 볼륨 가시화는 3차원, 혹은 그 이상의 차원의 공간에서 정의된 추상적이고 복잡한 볼륨 데이터로부터 의미 있고 가치적인 정보를 효과적으로 추출하도록 도와주는 다양한 기술에 관한 연구 분야로서, 기상학, 의학, 계산 유체 역학 등 여러 학문 분야에서 점차 그 중요성을 더해가고 있다. 한편 가상 현실은 컴퓨터가 만든 가상의 세상에 사용자가 몰입하여 시각, 청각, 촉각 등의 감각을 이용하여 세상을 경험하고 대화식으로 정보를 주고받을 수 있도록 도와주는 여러 기술에 관련된 연구 분야로서 국내외적으로 활발한 연구가 진행되고 있다.

본 논문에서는 CAVE라 불리는 몰입형 3차원 가상 환경 시스템 환경에서 인체 볼륨 데이터를 보다 개선된 볼륨 가시화 방법을 사용하여 실시간으로 입체 영상을 생성해주는 시스템을 설계하고 구현하였다. 이 시스템은 기존 3차원 텍스처 매핑 기반 볼륨 렌더링 방법의 느린 속도를 보완하고자, 영상 기반 렌더링에 기반을 둔 향상된 텍스처 매핑 기법을 사용하여 실시간 볼륨 입체 가시화 기능을 지원하며, 사용자를 위하여 다양한 인터페이스 기능들을 제공한다. 본 시스템의 효용성을 증명하기 위한 테스트 데이터로서 Visible Korean Human 데이터를 사용하였다. 본 논문에서는 실시간 입체 영상 시스템에 필요한 가시화 기법과 라이브러리, 그리고 구체적인 구현 내용에 대해서 설명한다.

키워드 : CAVE 시스템, 입체 영상, 볼륨 가시화, 영상 기반 렌더링, 텍스처 매핑

Abstract Volume visualization is an important subarea of scientific visualization, and is concerned with techniques that are effectively used in generating meaningful and visual information from abstract and complex volume datasets, defined in three- or higher-dimensional space. It has been increasingly important in various fields including meteorology, medical science, and computational fluid dynamics, and so on. On the other hand, virtual reality is a research field focusing on various techniques that aid gaining experiences in virtual worlds with visual, auditory and tactile senses.

In this paper, we have developed a visualization system for CAVE, an immersive 3D virtual environment system, which generates stereoscopic images from huge human volume datasets in real-time using an improved volume visualization technique. In order to complement the 3D texture-mapping based volume rendering methods, that easily slow down as data sizes increase, our system utilizes an image-based rendering technique to guarantee real-time performance. The system has been designed to offer a variety of user interface functionality for effective visualization. In this article, we present detailed description on our real-time stereoscopic visualization system, and show how the Visible Korean Human dataset is effectively visualized on CAVE.

· 본 연구는 한국과학기술정보연구원의 슈퍼컴퓨팅센터와 2001년도 서강대학교 교내 연구비의 지원을 받아 이루어졌음.

[†] 비회원 : 엔젠테크놀로지 연구원
classicwow@hanmail.net

^{**} 비회원 : 한국과학기술정보연구원 슈퍼컴퓨팅센터 연구원
jylee@hpcnet.ne.kr

^{***} 비회원 : 한국과학기술정보연구원 슈퍼컴퓨팅센터 실장

msjoh@hpcnet.ne.kr

^{****} 비회원 : 한국과학기술정보연구원 슈퍼컴퓨팅센터장
sslee@hpcnet.ne.kr

^{*****} 종신회원 : 서강대학교 컴퓨터학과 교수
ihm@sogang.ac.kr

논문접수 : 2002년 4월 11일

심사완료 : 2002년 8월 20일

Key words : CAVE system, stereoscopic image, volume visualization, image-based rendering, texture mapping

1. 서론

1.1 연구배경

컴퓨터 그래픽스는 컴퓨터를 이용하여 다양한 영상을 생성하는 것에 관련된 각종 기술을 연구·개발하는 분야로서, 지난 수십 년간 지속적인 기술 개발로 인하여 발전을 거듭하고 있다. 이러한 컴퓨터 그래픽스 기술에 기반을 두는 과학적 가시화(scientific visualization)의 한 분야인 볼륨 가시화(volume visualization)는 최근 활발히 연구되고 있는 분야 중의 하나로서, 3차원, 혹은 그 이상의 차원의 공간에서 정의된 복잡하고 추상적인 볼륨 데이터로부터 의미 있고 가시적인 정보를 추출하도록 도와주는 기술에 관련된 연구분야이다. 볼륨 데이터는 MRI, CT, SPECT 등의 장치를 사용하는 의료분야나 기상학, 계산 유체 역학, 물리, 화학 등의 다양한 분야에서 빈번히 산출되고 있으며, 날로 그 복잡도가 증가하고 있다.

한편 가상 현실(VR: Virtual Reality)은 사용자가 컴퓨터가 만들어낸 가상의 세계 속에 몰입하여, 직접 이동하거나 여러 각도로 사물을 보거나, 소리를 듣고, 사물을 만지는 등의 행위를 통하여 가상의 세상을 경험하게 해주는 컴퓨터 기술에 관하여 연구하는 분야이다. CAVE(Cave Automatic Virtual Environment) 시스템은 1991년 시카고 일리노이즈 대학의 EVL(Electronic Visualization Laboratory)에서 개발된 투영 기반 몰입형 가상 현실(projection-based immersive virtual reality) 시스템으로서, 1992년 SIGGRAPH 학회에서 처음 소개되면서 국제적으로 주목받게 되었다[1]. 이 CAVE 시스템은 컴퓨터 그래픽스 가시화 분야에서 혁신적인 도구로서 3m×3m×3m 크기 정도의 정육면체 모양을 하고 있으며, 뷰잉 도구인 3차원 안경(three-dimensional shutter glasses)과 완드(wand) 등의 3차원 인터페이스 도구를 통하여 사용자가 가상 환경에 실제로 존재하는 듯한 느낌을 갖게 해준다. 이 CAVE는 군사, 과학, 제조산업, 항공우주 그리고 의료 영상 등 다양한 분야에 활용될 수 있어서, 처음 나온 이후로 이것을 이용한 많은 연구가 국내외에서 활발히 이루어져왔으며, 앞으로도 많은 연구가 이루어질 것으로 예상된다.

1.2 관련 해외 연구

해외에서 개발한 CAVE 시스템 중 중요한 몇 가지를 살펴보면 다음과 같다. [2]에서는 CAVE 시스템 전반에

걸쳐 자세히 설명을 하고, 이러한 CAVE가 VR 환경에 필요한 많은 필요 조건들을 충분히 만족한다는 것을 실험적으로 증명해 보였다. 또한 [3]에서는 CAVE에서 장애자를 위한 인터페이스를 개발하였다. 특히 휠체어를 이용하는 사람들의 가상 환경을 위한 특별한 인터페이스를 개발하였다.

[4]에서는 SANDBOX(Scientists Accessing Necessary Data Based On eXperimentation)라는 과학적 데이터를 위한 새로운 인터페이스를 제안하였는데, 이는 연구원이 가상의 환경에서 실험을 할 수 있도록 도와준다. 즉, SANDBOX를 이용하면 연구원이 가상 환경에서 가상의 실험 도구들로 실제 실험을 그대로 재현할 수 있으며, 결과로 수집되는 데이터도 실제와 동일하게 얻어진다. 또한 고속 네트워크와 고성능 워크스테이션을 사용하여 CAVE에서 과학적 데이터를 가시화 해주는 분산 시스템도 개발되었다[5]. 이 연구에서는 기존 시스템에서 미리 생성해놓은 과학적 데이터만을 가시화 해주는 문제점을 극복하여 거의 실시간으로 복잡한 과학적 시뮬레이션 계산을 수행하면서 가시화 해주는 것을 가능케 하였다. 그러한 예로서 '우주 웜'(Cosmic Worm)이라는 과학적 어플리케이션을 소개하였다.

또한 CAVE 시스템 상에서 역학 시스템의 가상 프로토타입핑을 가능하게 해주는 소프트웨어가 개발되기도 하였다[6]. 이 소프트웨어는 역학 시스템을 분석하는 시뮬레이션 소프트웨어 FIFEA(Fast Implicit Finite Element Analysis) 부분과 결과를 가시화하는 가상 현실 디스플레이 소프트웨어 부분 두 가지 요소로 나뉘어 지는 데, 이 논문에서는 두 소프트웨어를 결합하는 데 있어 필요한 여러 가지 연구 문제에 대해 살펴보았다. 또한 CAVE 환경에서 어린이들을 위한 몰입형 교육 환경도 개발이 되었다[7]. 이는 EVL에서 NICE (Narrative Immersive Constructionist/Collaborative Environments)라는 프로젝트 하에 개발된 가상 교육 환경 시스템으로 어린이가 간단한 가상 생태계를 만들고 재배하는 기능을 제공하고, 네트워크를 통하여 다른 곳에 위치한 어린이와의 협동도 가능하게 하였다.

1.3 관련 국내 연구

국내에서는 아직까지 직접 CAVE 시스템을 위한 응용 소프트웨어의 개발이 외국처럼 활발하지는 못한 상태이다. 이는 CAVE 시스템이 국내에 도입된 지 얼마 되지 않았고, 아직 국내에서는 대전의 한국과학기술정보

연구원(KISTI, Korea Institute of Science and Technology and Information) 산하 기관인 슈퍼컴퓨팅 센터 등 극히 소수의 기관에 CAVE가 설치되어 있기 때문이다. CAVE와 유사한 VR 시스템으로는 한국과학기술연구원(KIST, Korea Institute of Science and Technology)의 '리얼리티 스튜디오'(Reality Studio)가 있다. 이 스튜디오는 18평 규모를 가지는 스튜디오로서 구의 일부분을 절단한 모양의 가로 9.55m, 세로 3.145m 크기의 상하좌우가 모두 곡면인 스크린을 구비하고 있다. 그리고 경주 불국사 석굴암 내부를 가상 체험할 수 있도록 재현한 콘텐츠가 시연되고 있다. 또한 이곳에는 자전거 시뮬레이터가 있는데 이것은 자전거의 핸들 부위와 페달 부위에 두 개의 모터를 달아 자전거를 주행하는 영상을 보면서 영상에 따라 두 모터에서 핸들의 위치와 페달의 이동 속도를 감지해 3차원의 시청각을 체험할 수 있게 해주는 시스템이다. 보고 듣는 환경에 따른 느낌을 회전력이 매우 큰 두 개의 모터를 이용해 핸들 및 페달에 느낌을 전달하도록 한 것이다. 이 자전거 시뮬레이터는 고정된 자전거에서 페달을 밟더라도 펼쳐지는 영상에서 보여지는 공간에서 자전거를 타는 것처럼 핸들과 페달의 힘이 실감나게 전달되기도 한다.

본 논문에서는 이와 같이 아직 우리 나라에서 연구가 미흡한 상태인 CAVE 시스템을 기반으로 하는 실시간 입체 영상 가시화 시스템을 개발하려 하였다. 특히 본 시스템에서는 테스트 데이터로 VKH(Visible Korean Human)을 사용한다. VKH은 한국과학기술정보연구원의 주관 하에 생성되고 있는 한국인 최초의 고해상도의 인체 영상 데이터이다[8]. 미국의 경우 NLM(National Library of Medicine)에서 이미 지난 86년부터 Visible Human 프로젝트를 기획하여, 지난 94년과 95년에 각각 백인 남성 및 여성에 대한 정밀 영상 데이터를 배포하였다[9]. 그러나 이 데이터는 인체 구조가 한국인 신체와는 달라 국내에서의 교육과 연구에 어려움이 많았다.

고해상도의 VKH 데이터는 그 크기가 매우 방대하여 기존의 3차원 텍스처 매핑 기법을 사용하여 가시화할 경우 속도가 매우 느려 실시간으로 가시화하기가 용이하지 않다. 따라서 본 시스템에서는 3차원 볼륨 데이터의 전처리를 통하여 생성한 중간 단계의 데이터를 2차원 텍스처 형태로 저장한 뒤, 렌더링 시 텍스처 메모리에 올려놓고 대화식으로, 그리고 실시간으로 가시화해주는 영상 기반 렌더링 기법을 도입하였다. 이 기법은 3차원 텍스처 매핑에 비해 속도가 훨씬 빠른 2차원 텍스처 매핑을 사용하여 구현하기 때문에 실시간 가시화가 가능하다. 또한 이 기법은 법선 벡터 데이터를 이미지로

저장하고 있어, 기존의 영상 기반 렌더링 기법들과는 달리 사용자가 실시간으로 광원이나 물질의 성질을 동적으로 바꿀 수가 있다는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 시스템을 개발하는 데 필요한 텍스처 매핑 기반 볼륨 렌더링에 대한 과거 연구를 간략히 설명하고 시스템의 실시간 가시화를 위하여 새롭게 개선된 영상 기반 텍스처 매핑 알고리즘에 대하여 알아보겠다. 또한 입체 영상을 위한 입체 영상 생성 기법에 대해서도 간략히 알아보겠다. 3장에서는 시스템에 사용된 하드웨어에 대하여 간략히 설명하고 실제 시스템의 구조에 대하여 설명하겠다. 또한 실제 시스템 구현 내용에 대해서도 설명하고, 마지막으로 4장에서는 본 연구의 결론을 내리고 앞으로의 연구 방향에 대해서 고찰한다.

2. 텍스처 매핑 기반 볼륨 렌더링

볼륨 데이터로부터 3차원 그래픽스 기법을 사용하여 이미지를 생성해주는 기법인 볼륨 렌더링에 있어서 가장 중요한 문제는 렌더링 속도와 결과 이미지의 사실감이라고 할 수 있다. 따라서 과거부터 가능한 한 사실적인 결과 이미지를 생성하면서 빠른 속도로 렌더링을 해주는 방법에 대한 연구가 활발히 이루어져왔다. 특히 90년대 초반 제안된 이후[10,11,12,13], 그래픽스 하드웨어의 기능 확장과 함께 발전해온, 3차원 텍스처 매핑 하드웨어를 기반으로 하는 볼륨 렌더링 기법은 고화질의 결과 이미지를 생성하면서도 실시간 렌더링이 가능하여 크게 주목을 받고 있다. 본 논문에서 개발한 시스템은 기존의 3차원 텍스처 매핑 기반 볼륨 렌더링 기법을 변형한 새로운 형태의 실시간 알고리즘을 사용한다. 본 장에서는 우선 본 논문에서 개발한 시스템의 기반이 되는 텍스처 매핑 기반 볼륨 렌더링 방법에 대하여 설명한 후, CAVE 시스템 상에서의 입체 가시화를 위한 입체 영상 생성 기법에 대해서 간략히 설명한다.

2.1 다단계 알고리즘에 기반을 둔 볼륨 렌더링 기법

본 논문에서는 SGI의 그래픽스 가속기를 효과적으로 이용하기 위하여 기존의 하드웨어 기반 볼륨 렌더링 방법보다 우수한 형태의 풍의 조명 모델에 기반을 둔 다단계 렌더링 알고리즘을 사용한다. 이 방법은 본 연구실에서 '프로그래밍 가능한 셰이더'(programmable shader) 기능이 제공되지 않은 가속기용으로 개발한 기법으로서, 크게 두 단계로 이루어져 있다. 우선 첫 번째 단계에서는 볼륨 데이터 공간의 각 지점에서의 법선 벡터를 저장하는 3차원 텍스처를 뷰잉 방향에 수직인 다각형 슬라이스들을 사용하여 합성한 2차원 형태의 법선

벡터 영상을 생성한다. 다음 두 번째 단계에서는 앞에서 만들어진 2차원 법선 영상에 앰비언트 반사와 난반사, 그리고 정반사 요소에 대한 계산을 2차원 텍스처 매핑 기법을 통하여 수행하여 최종 결과 영상을 생성한다. 여기에서 사용하는 풍의 조명 모델 식은 $\vec{C} = M_{ad} \cdot N + \epsilon(M_s \cdot N)^n$ 와 같으며, 이에 대한 행렬식은 다음과 같이 나타낼 수 있다.

$$\begin{pmatrix} C_x \\ C_y \\ C_z \\ 1 \end{pmatrix} = \begin{pmatrix} \delta_x L_x & \delta_y L_y & \delta_z L_z & \gamma_x \\ \delta_x L_x & \delta_y L_y & \delta_z L_z & \gamma_x \\ \delta_x L_x & \delta_y L_y & \delta_z L_z & \gamma_x \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} N_x \\ N_y \\ N_z \\ 1 \end{pmatrix} + \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ 1 \end{pmatrix} \begin{pmatrix} H_x & H_y & H_z & 0 \\ H_x & H_y & H_z & 0 \\ H_x & H_y & H_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} N_x \\ N_y \\ N_z \\ 1 \end{pmatrix}^{n_s}$$

본 논문의 가시화 소프트웨어에서 사용하는 렌더링 알고리즘을 요약하면 다음과 같다.

1. <단계 1>

3차원 텍스처 매핑을 통해 법선 벡터를 누적하여 합성한 뒤, 결과 영상으로 2차원 텍스처(N)를 만든다.

2. <단계 2>

(가) 색깔 행렬(color matrix) 기능을 사용하여 N과 앰비언트 반사와 난반사 요소를 적용하여 M_{ad} 텍스처를 만든다.

(나) N과 정반사 요소를 적용하여 M_s 텍스처를 만든다.

(다) 정반사 계수를 사용하여 ϵ 텍스처를 만든다.

(라) M_s 텍스처 이미지를 프레임 버퍼에 n_s 번 그려서 정반사 요소를 법선 벡터에 적용한 영상을 생성한다.

(마) ϵ 텍스처 이미지를 프레임 버퍼에 적절히 합성하여 정반사 요소의 계수를 적용한 영상을 생성한다.

(바) M_{ad} 텍스처 이미지를 프레임 버퍼에 적절히 합성

하여 앰비언트 반사와 난반사까지 적용한 영상을 생성한다.

이 방법의 <단계 1>은 3차원 텍스처 매핑 기법을 이용한, 삼선형 보간에 의한 3차원 볼륨 데이터의 리샘플링 과정에 해당하고, <단계 2>는 셰이딩 계산에 해당한다. 그림 1의 (a)부터 (i)까지는 본 렌더링 알고리즘의 각 단계가 수행된 직후 생성된 이미지를 도시하고 있다. 그림 (e), (f), (g)는 각각 n_s 가 2, 5, 10인 경우로서 여러 번 반복하여 그림으로써 정반사 효과가 더욱 부각되는 것을 알 수가 있다. 이 방법은 한 장의 이미지를 만들기 위하여 여러 번의 렌더링 패스를 통하여 생성한 이미지를 합성하여 최종 영상을 생성하는 전형적인 다단계 알고리즘이라 할 수 있다. 최근에 발표되고 있는 NVIDIA나 ATI 등의 그래픽스 가속기는 꼭지점 셰이더(vertex shader)와 픽셀 셰이더(pixel shader) 등과 같은 셰이더를 통하여 렌더링 파이프라인의 일부를 개발자가 프로그래밍할 수 있도록 해주는 기능을 제공한다. 이러한 그래픽스 가속기를 사용할 경우 다단계 알고리즘을 사용할 필요가 없이 한 번의 렌더링 패스를 통하여 원하는 이미지를 생성할 수가 있다. 그러나 본 논문에서 사용하는 SGI의 그래픽스 가속기에서는 셰이더 기능을 제공하지 않기 때문에 위에서와 같은 다단계 알고리즘을 사용하였다.

한 가지 주의할 점은 이 다단계 방법의 경우 <단계 1>에서 3차원 텍스처 매핑 기능을 사용하기 때문에 이 부분이 전체 렌더링 시간의 대부분을 차지한다. 특히 본



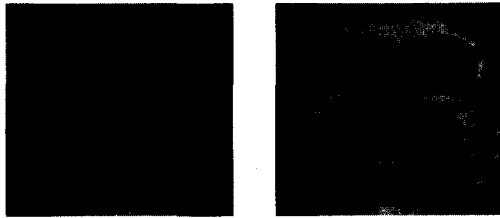
(a) <단계 1> (b) <단계 2> (가) (c) <단계 2> (나) (d) <단계 2> (다)



(e) <단계 2> (라)-1 (f) <단계 2> (라)-2 (g) <단계 2> (라)-3 (h) <단계 2> (마) (i) <단계 2> (바)

그림 1 다단계 알고리즘의 계산 과정

연구에 시도하려는 것과 같이 데이터의 크기가 방대한 경우 <단계 1>의 계산으로 인하여 실시간 렌더링을 위한 충분한 속도를 기대하기 어렵다. 따라서 본 논문에서는 다음절에 설명하는 바와 같이 이 다단계 기법에 영상 기반 렌더링을 적용시킨 방법을 사용하여 실시간으로 방대한 데이터를 가시화하려 하였다.



(a) 광선 추적법 사용 (b) 본 논문의 제시 방법 사용

그림 2 광선 추적법 결과와의 화질 비교

그림 2는 본 논문에서 제시한 기법을 사용하여 생성한 이미지와 가장 우수한 화질을 생성해주는 광선 추적법을 사용하여 얻은 이미지를 비교하고 있다. 여기서 광선 추적법 이미지를 생성할 때 가능한 한 텍스처 기반 방법과 동일한 상황의 렌더링 인자를 사용하도록 하였다. 두 방법의 리샘플링 방식이나 렌더링 인자의 차이로 인하여 결과 형태에 약간의 차이가 있음을 볼 수가 있으나, 전체적인 화질 면에서 본 논문에서 사용한 텍스처 기반 방법이 광선 추적법에 근접하는 우수한 화질의 이미지를 생성함을 알 수가 있다.

2.2 영상 기반 볼륨 렌더링

앞에서 언급한 3차원 텍스처 매핑 하드웨어를 이용한 볼륨 렌더링 기법은 고품질의 결과 이미지와 빠른 렌더링 속도를 가능케 하지만, 이러한 3차원 텍스처 매핑 기법 역시 아직까지 광선추적법과 같은 기법에 비해서는 화질이 떨어지고, 렌더링 속도 역시 대용량 볼륨 데이터를 렌더링할 때에는 대화식 가시화 소프트웨어 개발에 충분하지 못하다는 문제가 있다. 이러한 문제를 해결하고자 본 연구에서는 앞에서 설명한 알고리즘에서 3차원 텍스처 매핑 계산에 해당하는 <단계 1>과 나머지 부분을 분리를 하였다. 즉 렌더링에 대부분의 시간을 차지하는 3차원 리샘플링 계산을 하드웨어가 아니라 광선 추적법을 통하여 고품질의 법선 영상을 미리 만들어 놓고, 실제 가시화 작업 시에는 3차원 텍스처 매핑 기법을 사용하지 않고 빠른 2차원 텍스처 매핑 기능만을 사용하기 때문에 빠른 속도로 최종 결과 이미지를 생성할 수 있도록 하였다.

다시 말해서 본 논문에서는 앞에서 설명한 다단계 알고리즘 중 <단계 1> 부분이 렌더링 시간의 대부분을 차지하고, 또한 법선 정보만 있으면 풍의 조명 모델을 적용할 수 있다는 점을 고려한 영상 기반 렌더링 방법을 사용한다. 즉 다양한 시점에 대하여 <단계 1>까지의 결과 영상, 즉 법선 정보를 미리 생성하여 2차원 텍스처 형태로 저장을 한 후, 실제 렌더링 시에는 빠른 속도로 계산이 가능한 <단계 2>부분만을 수행시켜 대화식으로 실시간 영상을 생성한다. 이 방법은 전처리된 영상을 사용함으로써 계산 시간을 많이 줄여 방대한 크기를 가지는 볼륨 데이터의 실시간 렌더링을 가능케 해주며, 또한 기존의 영상 기반 기법과는 달리 사용자가 대화식으로 광원과 물체에 대한 조사를 가능하게 해준다. 특히 <단계 1>의 계산은 좋은 화질의 영상을 생성하기 위하여 전처리 단계에서 기존의 볼륨 데이터를 위한 광선 추적법[15, 16]을 변형하여 법선 데이터를 합성하였다. 원래의 광선 추적법 알고리즘에서 광선의 각 리샘플링 지점에서 주변 복셀의 밀도 값을 사용하여 현 위치의 그래디언트 방향을 구한 후 이를 법선 벡터 값으로 사용하여 셰이딩 알고리즘을 적용시켜 색상 값을 구한다. 그러나 본 논문에서는 법선 벡터 정보를 필요로 하기 때문에 셰이딩 알고리즘을 적용시키기 전 단계의 그래디언트 값만을 합성하여 법선 벡터 영상을 생성한다. 이렇게 만들어진 영상을 NTEX(Normal TEXTure) 확장자를 가지는 파일에 저장한 후, 렌더링 시 텍스처 메모리에 올려 사용한다. 또한, 본 논문에서는 영상 평면의 한 점에서 한 개의 광선을 쏘는 일반 광선 추적법이 아닌, 여러 개의 광선을 쏘는 슈퍼 샘플링 방법을 사용하여 렌더링함으로써 보다 나은 화질의 법선 벡터 영상을 얻을 수 있었다.

영상 기반 렌더링 방법은 전통적인 렌더링 방법들이 전경을 표현해주는 기하 모델 정보를 이용하여 영상을 생성하는 것과는 달리 미리 준비된 영상을 적절히 처리하여 렌더링된 최종 영상을 생성한다. 주로 실제로 촬영한 사진이나 미리 합성한 이미지를 사용하는데, 만일 실제의 사진을 이용한다면 기존의 방법들이 가지고 있는 문제중의 하나인 물체를 모델링하고 전경에 배치하는 힘든 작업을 수월하게 할 수 있다. 이 방법의 장점은 계산량이 적고, 전경의 복잡도와는 상관없으며, 입력 이미지로 실제 사진이나 모델들을 렌더링한 사진들도 가능하다는 것이다. 그러나 이러한 영상 기반 렌더링 기법은 종종 렌더링 인자들이 변할 경우 입력 이미지를 다시 만들어야 한다는 치명적인 결점을 갖고 있다. 본 연구에서 이와 관련된 렌더링 인자 중 대표적인 것으로 광원

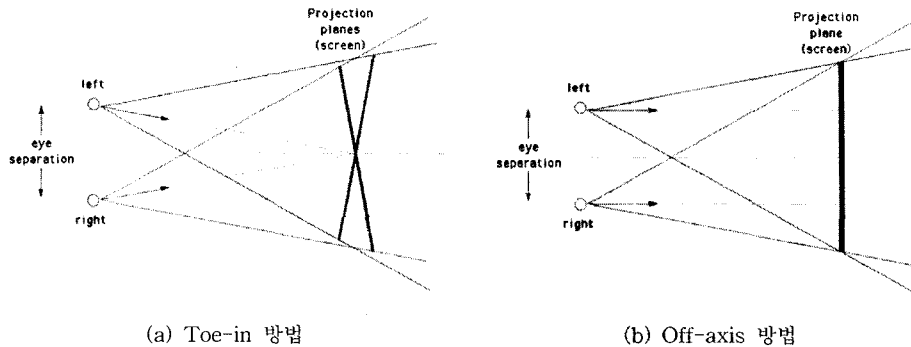


그림 3 입체 영상 생성 방법[17]

의 위치와 색깔, 그리고 물체의 색깔 등을 들 수가 있다. 본 논문에서 제안하는 가시화 방법은 최종 결과 이미지를 미리 만들어 놓는 것이 아니라, 각 시점에 해당하는 볼륨 데이터의 법선 데이터를 만들어 놓기 때문에, 위와 같은 렌더링 인자들이 변할 때에도 고비용의 3차원 리샘플링 계산을 통하여 다시 이미지를 재구성할 필요가 없이 <단계 2>에 해당하는 간단한 셰이딩 계산을 통하여 그러한 변화를 반영할 수 있다.

2.3 입체 영상의 생성

2.3.1 입체 영상 생성 기법

본 논문에서 구현한 가시화 시스템은 CAVE 상에서 입체적인 영상을 생성하는 것을 목표로 한다. 입체 영상(stereoscopic image)은 좌우 두 눈의 위치에서 각각 대상을 렌더링한 후 동시에 디스플레이 함으로써 생성할 수 있다. 각 눈에 해당하는 영상은 모니터에 디스플레이 한 후 특수 안경을 사용하여 보면 안경을 통하여 각각의 눈 위치에 해당하는 곳에서 렌더링한 영상이 두 눈으로 들어옴으로써 입체감을 느끼게 된다. 이렇게 입체감을 느끼는 가장 큰 이유는 인체의 두 눈이 가로 방향으로 약 65mm 정도 떨어져서 존재하기 때문에 나타나는 양안시차(binocular disparity) 때문이다. 즉, 좌우의 눈은 각각 서로 다른 2차원 영상을 보게 되고, 이 두 영상이 망막을 통해 뇌로 전달되면 뇌는 이를 정확히 서로 융합하여 본래 3차원 영상은 깊이감과 실제감을 재생하게 된다.

입체 영상을 생성하는 대표적인 방법으로 크게 Toe-in 방법과 Off-axis 방법 두 가지 방법으로 나누어진다 [17]. Toe-in 방법은 그림 3(a)에서 보면 알 수 있듯이 두 시점에 해당하는 카메라가 서로 같은 초점(focal point)을 사용하는 방법이다. 그러나 이 방법은 여러 왜곡이 일어날 수 있어서 아주 이상적인 방법은 아니다.

특히 영상의 중앙 부분에서 수직시차(vertical parallax)가 증가해서 일어나는 왜곡이 대표적인 왜곡의 예이다. 반면 Off-axis 방법은 그림 3(b)에서처럼 카메라의 방향을 평행하게 하는 방법으로서, Toe-in 방법보다 조금 더 과학적으로 정확한 방법으로, 수직시차가 발생하지 않기 때문에 눈이 덜 피로하다. 그러나 이 방법은 특별히 뷰잉 프러스텀(viewing frustum)을 조정해줘야 한다는 단점이 있다.

2.3.2 입체 영상 생성 기법의 적용

본 시스템에서는 Toe-in 방법과 Off-axis 방법 모두를 구현하였다. 기본적으로 시점이 그림 4에서와 같이 렌더링할 대상의 주위를 원을 그리면서 이동하여 미리 영상을 저장해 놓는다. 그래서 Toe-in 방법일 경우 미리 입체감이 잘 느껴지는 적당한 좌우 각도를 실험적으로 찾아 렌더링 시 그 각도만큼 떨어진 텍스처를 읽어와 좌우 눈 위치에 해당하는 영상으로 렌더링되도록 구현할 수 있다. 그러나 Off-axis 방법일 경우 뷰잉 방향이 평행해야 하므로 그림 3(a)에서와 같은 뷰잉 방향을 사용할 수 없으므로, 각 시점 위치에서 따로 좌우 영상을 계산하여 저장해 놓는다. 그래서 실제 렌더링 시 Toe-in 방법에 비해 두 배의 텍스처 메모리가 요구된다. 그러나 결과 비교 시 Off-axis 방법이 Toe-in 방법보다 크게 뛰어난 편이 아니어서 본 시스템에서는 메모리를 효율적으로 사용하기 위해서 Toe-in 방법을 주로 사용하였다. 이 경우 각 시점마다 좌우 영상을 만들 필요 없이 적당한 각도 차의 두 영상을 좌우 영상으로 사용하게 되면, 메모리가 절반만 필요하게 되어 보다 효율적이다. 예를 들어, 그림 4에서 볼 때 3번 위치에 시점이 위치할 경우 4번을 좌측 영상, 2번을 우측 영상으로 사용할 수 있도록 하였다.

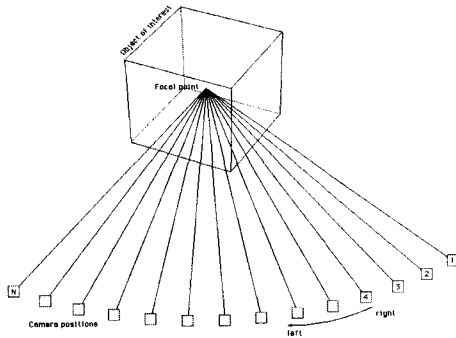


그림 4 입체 영상 생성[18]

3. 시스템 설계 및 구현

대화식의 입체 영상 생성 시스템을 구현하기 위해서는 매 순간마다 좌우 두 장의 이미지를 생성해야 하기 때문에 무엇보다도 실시간으로 입체 영상을 생성해주는 볼륨 데이터 가시화 기법의 적용이 필수적이다. 또한 이 시스템을 사용하는 사용자가 CAVE 시스템 상에서 쉽고 편리하게 주어진 기능들을 이용할 수 있도록 해주는 효과적인 사용자 인터페이스를 개발하는 것도 중요하다고 할 수 있다. 본 연구에서는 이러한 사항들을 고려하여 방대한 볼륨 데이터의 입체 영상 가시화 시스템을 개발하였는데, 핵심 렌더링 기법은 C/C++와 OpenGL을 사용하여, 그리고 병렬 그래픽스 시스템의 구동 및 인터페이스 부분은 SGI OpenGL Performer를 사용하여 구현하였다. 또한 CAVE 시스템 상에서 사용자의 입력을 3차원 인터페이스 도구인 완드를 통하여 자유롭게 받을 수 있도록 TrackdAPI라는 라이브러리를 사용하였다.

본 시스템의 전체 구조는 그림 5에 도시한 바와 같이 크게 소프트웨어 부분인 렌더링 시스템과 하드웨어 부분인 CAVE 시스템으로 나눌 수 있는데, 후자인 CAVE 시스템은 컴퓨팅 시스템인 Onyx 3400과 디스플레이 시스템인 리액터(ReaCTor)로 구성되어있다. 그리고 렌더링 시스템은 크게 렌더링 엔진 부분과 사용자 인터페이스로 나누어진다. 본 장에서는 먼저 전체 시스템의 기반이 되는 CAVE 시스템에 대하여 설명한 후, 핵심 부분인 렌더링 시스템 부분에 대해 설명하겠다.

3.1 CAVE 시스템

전기한 바와 같이 CAVE 시스템은 보통 3m×3m 정도 크기의 면 다섯 개로 이루어진 정육면체 모양을 하고 있으며, 각 면은 각각 하나씩의 고해상도 프로젝터와 연결되어 있다. 그리고 입체 안경을 통한 입체 투영을 제공하는데, 입체 안경이 여러 개가 있어서 동시에 여러

명의 사용자가 동일한 가상 환경을 체험할 수 있도록 해준다. 본 논문에서 개발한 입체 영상 시스템이 실행되는 한국과학기술정보연구원의 CAVE 시스템 SeeMore의 경우 디스플레이 시스템인 리액터(ReaCTor)와 이러한 리액터의 각 면의 디스플레이를 담당하는 다섯 개의 그래픽스 파이프라인들로 구성된 SGI Onyx 3400 가시화 시스템으로 나누어진다(그림 5).

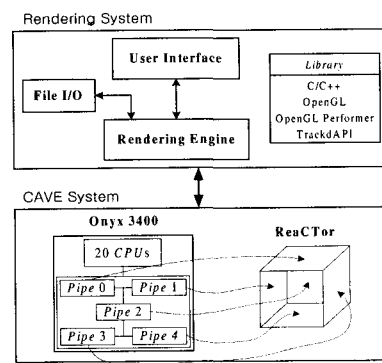


그림 5 가시화 시스템 구조

3.1.1 리액터(ReaCTor)

SeeMore 시스템의 디스플레이를 담당하는 리액터는 Trimension사의 입방체 디스플레이 시스템으로서 방모양의 여러 개의 스크린으로 이루어져 있으며 최대 6개까지 스크린을 지원한다[20]. 또한 각각의 스크린은 고휘도 및 고해상도를 지원하는 CRT 프로젝터로 연결되어 있어서 여러 명의 사용자가 동시에 몰입 가상 환경을 체험할 수 있도록 해준다. 이때 보조 장비로 입체 안경인 LCD shutter glasses, 3D 인터페이스 도구인 완드, 무선 마이크 등이 사용된다. 또한 3D 오디오 시스템이 설치되어 있어서 시각적인 효과뿐만 아니라 청각적으로도 입체 음향을 제공하므로 사용자가 보다 완벽한 몰입 가상 환경을 체험할 수 있도록 도와준다.

3.1.2 SGI Onyx 3400

SGI Onyx 3400은 SGI 가시화 시스템인 Onyx 3000 계열의 시스템이다[21]. 본 논문에서 기반으로 하는 Onyx 3400 가시화 시스템은 최대 32개의 CPU, 64GB 주 메모리를 가질 수 있으며, 또한 최대 8개의 그래픽 하드웨어 파이프라인을 지원하여 동시에 8명이 그래픽 작업을 수행할 수 있다. 실제로 입체 영상 시스템이 사용하는 SeeMore 시스템의 경우 400MHz의 MIPS R12000 CPU 20개와 6GB의 주 메모리, 72GB의 내장

디스크, 그리고 다섯 개의 InfiniteReality3 그래픽스 파이프라인으로 구성되어 있다. 또한 각 그래픽스 파이프라인은 각각 160MB 프레임 버퍼와 256MB 텍스처 메모리를 가진다.

3.2 렌더링 시스템 구현

렌더링 시스템 부분은 전기한 바와 같이 가시화의 핵심 부분을 담당하는 렌더링 엔진과 사용자의 입력을 처리하여 주는 사용자 인터페이스로 나누어진다.

3.2.1 렌더링 엔진

C/C++과 OpenGL을 사용하여 구현한 렌더링 엔진은 우선 각 시점에 해당하는 법선 영상 데이터인, 확장자가 NTEX인 파일로 저장된 VKH 데이터를 읽어와 2차원 텍스처 형태로 정의하여 텍스처 메모리에 저장한다. 그런 후 사용자의 시점이 결정이 되었을 때, 콜백(callback) 함수 내에서 셰이딩 함수를 호출하여 2장에서 설명한 다단계 알고리즘의 <단계 2>의 계산을 통하여 결과 영상을 생성한다. 이때 좌우 두 눈에 해당하는 영상을 만들기 위하여 그에 해당하는 텍스처 이미지에 각각에 대하여 셰이딩 함수를 호출하는데, 함수 매개 변수로 현재 시점 위치에 대하여 사용할 텍스처의 인덱스 값과 어느 쪽 눈인지를 결정하는 플래그를 넘겨준다. 이 경우 셰이딩 함수 내에서는 뒤에서 설명할 물체 조각 모드에 따라서 피부에 해당하는 텍스처와 뼈에 해당하는 텍스처 중 적절히 선택하여 그린 후 엠비언트와 난반사 그리고 정반사 요소에 대한 이차원 텍스처를 적용하여 최종 원하는 영상을 생성하게 된다.

이러한 렌더링 엔진은 각각의 그래픽스 파이프마다 정의할 수 있기 때문에, 리액터의 각 면마다 각각 독립적으로 렌더링 엔진을 구동시킬 수 있다. 이 부분은 OpenGL Performer를 사용하여 구현하였는데, 아래는 코드는 구현된 프로그램의 일부분으로서 0번과 1번에 해당하는 파이프를 각각 리액터의 스크린 0번과 1번에 연결시키고, 또한 채널(channel) 0번과 1번에도 연결시킨 예를 보여준다.

```
/* set screen of pipe */
pipe[0] = pfGetPipe(0);
pfPipeScreen(pipe[0], 0);
Shared->pw[0] = pfNewPWin(pipe[0]);

pipe[1] = pfGetPipe(1);
pfPipeScreen(pipe[1], 1);
Shared->pw[1] = pfNewPWin(pipe[1]);

/* set channel of pipe */
pipe[0] = pfGetPipe(0);
```

```
chans[0] = pfNewChan(pipe[0]);
pipe[1] = pfGetPipe(1);
chans[1] = pfNewChan(pipe[1]);
```

이러한 방식으로 구현한 렌더링 엔진은 영상 기반 2차원 텍스처 매핑 기법을 사용함으로써 실행 중에 3차원 텍스처 매핑 부분까지 모두 계산을 하는 기존의 볼륨 렌더링 방법에 비해 월등한 시간 성능을 가진다. 표 1은 이러한 렌더링 속도를 비교하기 위하여 앞에서 설명한 다단계 알고리즘의 <단계 1>과 <단계 2>의 과정을 모두 수행하여 VKH의 상체 부분(데이터 해상도 512×512×512)의 피부를 렌더링한 시간(SKIN(3D texture))과 법선 정보로 텍스처 메모리에 올린 후 <단계 2>의 계산만 수행하여 결과 영상을 생성한 시간(SKIN(2D texture))을 보여주고 있다. 이 시간들은 한 개의 그래픽스 파이프를 사용하여 VKH를 1도씩 돌려가면서 측정한 평균 렌더링 시간인데, 기존의 3차원 텍스처 기반 볼륨 렌더링 알고리즘들이 보통 <단계 1>과 <단계 2>에 해당하는 정도의 계산을 수행해야하기 때문에, 전자(SKIN(3D texture))는 영상 기반 기법을 사용하지 않는 경우의 보편적인 3차원 텍스처 기반 볼륨 렌더링의 속도라 할 수 있다. 이 측정 결과에서 보면 알 수가 있듯이 본 논문에서 사용한 렌더링 기법은 기존 3차원 텍스처 매핑 기법을 사용했을 때 보다 약 24배 이상의 속도를 보여주고 있다. 그 결과 초당 약 45프레임의 결과 영상을 만들어 실시간 렌더링이 가능하였고 또한, 입체 영상을 생성할 때에도(SKIN(2D texture stereo)) 초당 약 23 프레임 정도의 속도로 렌더링을 할 수 있어, 본 논문에서 사용한 렌더링 방법은 실시간 대화식 입체 영상 생성에 사용하기에 충분한 성능을 보임을 알 수가 있다. 다시 한번 강조하면 최종 렌더링 결과물 영상 데이터로 사용하는 기존의 영상 기반 렌더링 방법과는 달리, 본 방법은 렌더링 과정의 앞부분(시간이 가장 많이 걸리는)만 계산한 일종의 중간 단계 이미지인 법선 영상을 영상 데이터로 사용하기 때문에 광원의 위치, 색깔, 물질의 색깔 등의 렌더링 인자들을 사용자가 대화식으로 동적으로 조절할 수가 있다.

표 1 렌더링 속도 비교

	속도(초)	프레임(fps)
SKIN(3D texture)	0.529	1.89
SKIN(2D texture)	0.022	45.62
SKIN(2D texture stereo)	0.044	22.81

3.2.2 사용자 인터페이스

전기한 바와 같이 본 연구에서 개발한 시스템은 사용자가 다양한 기능들을 편리하게 사용할 수 있게 하기 위하여 효과적인 인터페이스를 제공하도록 설계하였다. 일반 SGI 워크스테이션 환경 하에서 본 시스템은 간단한 마우스와 키보드 조작을 통하여 VKH 데이터를 원하는 각도에서 가시화할 수 있으며, 또한 광원, 물질 등을 자유롭게 조작할 수 있다. 전체적인 인터페이스 구현은 OpenGL Performer를 통하여 이루어졌으나, CAVE 환경 하에서는 풀 스크린 모드로 영상을 디스플레이 해 주는데, 이때 3차원 인터페이스 도구인 완드를 통하여 사용자의 입력을 받아들이고 또한 메뉴를 구현하기 위하여 TrackdAPI라는 라이브러리를 추가로 사용하였다. TrackdAPI는 VRCO(Virtual Reality Software and Consulting)에서 개발한 트래커(tracker)와 컨트롤러로부터 Trackd라는 데몬(daemon)을 통하여 공유 메모리에 저장된 값들을 읽어올 수 있게 해주는 라이브러이다[22].

본 시스템의 인터페이스는 크게 네 부분으로 나누어진다(표 2 참조). 첫째, 뷰 조작 기능은 사용자가 시점과 광원의 위치를 마우스 또는 완드를 통하여 자유롭게 조

작할 수 있게 해준다. 둘째, 물체 조작 기능은 VKH 데이터를 여러 가지 형태로 볼 수 있게 해주고, 색깔 조작 기능은 물체의 물질과 광원의 색을 자유롭게 바꾸는 기능을 제공하며, 마지막의 기타 조작 기능은 그 외의 가시화 관련 기능을 제공한다.

우선 뷰 조작 기능을 통하여 사용자는 마우스 또는 완드를 이용하여 데이터를 회전시킬 때, 시점과 광원 중 어느 것을 회전시킬지를 결정할 수 있다. 즉 사용자는 키보드의 '1', '2', '3' 등의 키를 사용하여 각각 HEAD_LIGHT, VIEW_ONLY, 그리고 LIGHT_ONLY 세 가지 모드로 변경할 수 있다. VIEW_ONLY과 LIGHT_ONLY 모드는 글자 그대로 각각 시점과 광원만을 회전시키는 모드이다. 그리고 HEAD_LIGHT 모드는 시점과 광원을 동시에 회전시키는 모드로 항상 바라보는 부분에 빛이 비춰진다. 이 세 가지 모두는 모두 기본적으로 시점의 위치에서 데이터를 가시화하고, 광원의 위치는 시점의 위치에 대하여 상대적으로 결정된다. 그림 6은 각각 이 세 가지 모드에서 렌더링한 결과를 보여준다. (a)는 바라보는 시점과 같은 곳에 광원이 존재하고, (b)와 (c)는 (a)에서 각각 광원과 시점만을 시계 반대방향으로 회전시켜 렌더링한 영상이다.

표 2 사용자 인터페이스 기능 요약

조 작	키(key)	기 능
뷰 조작	'1'	시점과 광원을 같이 이동 (HEAD_LIGHT 모드)
	'2'	시점만을 이동 (VIEW_ONLY 모드)
	'3'	광원만을 이동 (LIGHT_ONLY 모드)
물체 조작	'4'	피부 데이터 가시화 (SKIN 모드)
	'5'	뼈 데이터 가시화 (BONE 모드)
	'6'	피부 데이터에 미리 정의된 텍스처 입힘 (TEXTURED_SKIN 모드)
	'7'	뼈 데이터에 미리 정의된 텍스처 입힘 (TEXTURED_BONE 모드)
색깔 조작	'8'	피부, 뼈 데이터 동시 가시화 (SKIN_BONE 모드)
	'c'	LIGHT, SKIN_MATERIAL, BONE_MATERIAL, OFF 모드로 변환
	'r' 또는 'R'	red 요소를 감소, 증가
	'g' 또는 'G'	green 요소를 감소, 증가
	'b' 또는 'B'	blue 요소를 감소, 증가
	'l'	미리 정의된 광원 색깔 적용
	'm'	미리 정의된 물질 색깔 적용
기타 조작	'd'	미리 정의된 광원, 물질 색깔 적용
	's'	스테레오 뷰 제공
	'w'	현재 화면을 파일로 저장
	'+'	SKIN_BONE 모드 시 피부 투명도 증가
'-'	SKIN_BONE 모드 시 피부 투명도 감소	



(a) 헤드라이트 (b) 광원의 변화 (c) 시점의 변화

그림 6 뷰 조작을 통한 가시화 예



(a) SKIN 모드 (b) BONE 모드 (c) TEXTURED_SKIN 모드 (d) TEXTURED_BONE 모드 (e) SKIN_BONE 모드

그림 7 물체 조작을 통한 물질의 선택

물체 조작을 위하여 본 시스템은 입력 데이터로서 VKH 데이터에서 광선 추적법으로 생성한 피부와 뼈에 대한 법선 데이터를 사용한다. 따라서 어떤 데이터를 화면에 가시화할 지를 키보드 '4'-'8' 키를 누름으로써 결정할 수 있다. 각 키는 SKIN, BONE, TEXTURED_SKIN, TEXTURED_BONE, 그리고 마지막으로 SKIN_BONE 모드에 대응이 되는데, SKIN과 BONE 모드는 그 이름이 의미하는 바와 같이 각각 피부와 뼈를 화면에 가시화 해준다. 그리고 TEXTURED_SKIN과 TEXTURED_BONE은 위의 SKIN과 BONE 모드에서 가시화가 되는 부분에 자신이 따로 지정한 512x512 크기의 텍스처 이미지의 합성을 통하여 입혀줄 수 있는 기능을 가진 모드이다. 마지막으로 SKIN_BONE 모드에서는 피부 데이터의 투명도를 조절하여 피부와 뼈를 동시에 렌더링함으로써 피부 속의 뼈가 비추도록 가시화를 해준다. 그림 7은 위에서 설명한 다섯 가지 모드에서 렌더링한 결과를 보여준다.

다음 색깔 조작 기능은 'c' 키를 누름으로써 시작이 된다. 또한 이 'c' 키를 계속 누름으로써 LIGHT, SKIN_MATERIAL, BONE_MATERIAL, OFF 등 4가지 모드로 순환 변경이 가능한데, 순서대로 광원과 피부, 뼈 물질의 색을 바꿀 수 있는 모드이고 OFF는 색의 조절을 할 수 없는 모드이다. 색의 조절을 위하여 'r', 'R', 'g', 'G', 'b', 'B' 여섯 개의 키를 사용하여 RGB 요소들을 각각 0에서 255사이에서 바꿀 수 있다. 소문자는 그 요소의 값을 감소시키고, 반대로 대문자는 그 값을 증가

시킨다. 이 때 감소, 증가시키는 양은 정의한 값을 바꿈으로서 쉽게 바꿀 수 있다. 그리고 미리 정의되어 있는 색상도 있는데 LIGHT 모드에서 'l' 키를 누르면 미리 정의된 값으로 광원을 설정할 수 있다. 또한 SKIN_MATERIAL 모드에서는 'm' 키를 누르면 피부의 물질 값이 미리 정의된 값으로 설정되고, BONE_MATERIAL 모드에서는 뼈의 물질 값이 미리 정의된



(a) LIGHT 모드의 조작 예



(b) SKIN_MATERIAL 모드의 조작 예



(c) BONE_MATERIAL 모드의 조작 예

그림 8 색깔 조작을 통한 가시화 예

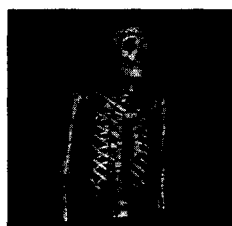
값으로 설정된다. 그리고 'd' 키를 누르면 광원과 피부, 뼈 물질 세 가지 모두가 미리 정의된 값으로 바뀌게 된다. 그림 8의 (a), (b), (c)는 각각 광원, 피부 물질, 뼈 물질의 색을 여러 가지로 바꾸어 그 결과를 렌더링한 예를 보여주고 있다.

기타 조작 기능은 위 세 가지 조작 기능 외에도 추가로 가시화 작업에 관련된 조작 기능을 제공한다. 우선 스테레오 버퍼(stereo buffer)를 사용하여 입체 영상을 생성하게 해주는 스테레오 기능이 있다. 이 기능은 키보드 's' 키를 누름으로써 실행되도록 구현하였다. 그러나 만약 구동하는 하드웨어의 그래픽 보드가 스테레오 버퍼를 지원하지 않을 경우 결과 이미지가 이상하게 나올 수 있다.

위의 스테레오 기능 외에 본 시스템은 'w' 키를 통하여 화면에 디스플레이 되는 영상을 'PPM' 파일 형식으로 저장할 수 있다. 또한 스테레오가 지원되는 하드웨어 일 경우에는 's' 키를 눌러 영상이 스테레오로 가시화 되 있을 때, 'w' 키를 눌러 파일을 저장하면 'Interlaced Stereo' 형식의 입체 영상으로 파일을 저장할 수 있다. 이 'Interlaced Stereo' 형식의 이미지는 좌우 영상을 한 줄씩 번갈아 가면서 저장하는 방법으로 일반 모니터에서도 간단한 장치를 설치함으로써 영상을 입체적으로 볼 수 있도록 해준다. 마지막으로 물체 조작 시 SKIN_BONE 모드에서 '+'와 '-' 키를 사용하면 피부 데이터의 투명도 값을 동적으로 조절할 수 있다(그림 9).



(a) 불투명도 0.6 (b) 불투명도 0.4



(c) 불투명도: 0.2

그림 9 기타 조작을 통한 피부 불투명도 변화 예

4. 결론

가상 현실은 최근에 활발히 연구되고 있는 컴퓨터 그래픽스의 응용 분야 중의 하나이다. CAVE 시스템은 이러한 가상 현실의 대표적인 시스템으로 몰입형 투영 기반 가상 환경을 제공한다. 해외에서는 과거부터 CAVE 시스템을 이용한 많은 연구가 이루어져 왔으나, 아직 국내에서는 CAVE 시스템이 보편화되지 않은 상태여서 활발한 연구가 이뤄지지 못했던 것이 사실이었다. 본 논문에서는 CAVE 시스템 상에서 방대한 의료 데이터를 대화식으로 가시화해주는 실시간 입체 영상 가시화 시스템을 개발하였다. 본 시스템은 기존의 3차원 텍스처 매핑 기법과 영상 기반 2차원 텍스처 매핑 방법을 혼합한 렌더링 방법에 기반하여 입체 영상을 실시간으로 생성할 수가 있다. 또한 렌더링의 기반이 되는 영상 데이터로 범선 정보에 해당하는 영상을 사용하기 때문에 기존의 일반적인 영상 기반 렌더링과는 달리 광원과 물질 등에 대한 동적 조정이 가능하다는 특징이 있다. 그림 10은 본 연구에서 개발한 가시화 소프트웨어를 사용하여 한국과학기술정보연구원의 CAVE 상에서 VKH 데이터를 대화식으로 조작하고 있는 모습을 보여주고 있다(CAVE의 각 면을 보여주기 위하여 과다 노출을 하였음.). 한편 그림 11은 서로 다른 물질 인자를 사용하여 생성한 두 영상을 각각 두 면에 도시하면서 가시화

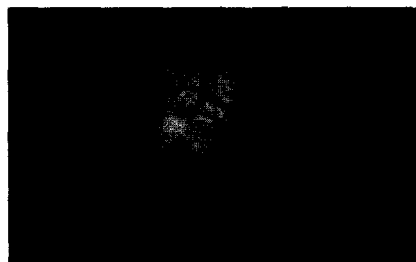


그림 10 CAVE 상에서의 조작 예 1

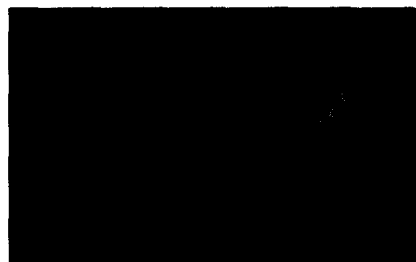


그림 11 CAVE 상에서의 조작 예 2

작업을 하는 모습을 보여주고 있다. 현재 이 시스템은 CAVE의 각 면에 대하여 그래픽스 파이프를 하나씩 연결을 하였기 때문에 특정 면에 대한 이미지를 만들기 위하여 시스템에 장착되어 있는 다섯 개의 파이프를 병렬적으로 사용할 수가 없다. 향후 본 시스템에 그러한 병렬 처리를 가능하게 해주는 하드웨어가 장착이 될 경우 보다 효율적으로 다섯 개의 파이프를 이용한 가상화 기법을 개발하게 될 것이다.

참 고 문 헌

- [1] C. Cruz-Neira, D. Sandin, T. DeFanti, R. Kenyon, and J. Hart, "The CAVE: Audio Visual Experience Automatic Virtual Environment," In Proceedings of ACM SIGGRAPH '92, pp. 65-72, June 1992.
- [2] C. Cruz-Neira, D. Sandin, and T. DeFanti, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," In Proceedings of ACM SIGGRAPH '93, pp. 135-142, August 1993.
- [3] D. Browning, C. Cruz-Neira, D. Sandin, T. DeFanti, and J. Edel, "Input Interfacing to the CAVE by Persons with Disabilities," In Proceedings of the Second Annual International Conference on Virtual Reality and People with Disabilities, pp. 1-9, San Francisco, CA, June 1994.
- [4] A. Johnson and F. Fotouhi, "The SANDBOX: A Virtual Reality Interface to Scientific Databases," In Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management, pp. 12-21, Charlottesville, VA, September 1994.
- [5] T. Roy, C. Cruz-Neira, D. Sandin, and T. DeFanti, "The Cosmic Worm in the CAVE: Steering a High Performance Computing Application from a Virtual Environment," Presence: Teleoperators and Virtual Environments, Vol. 4, No. 2, pp. 121-129, April 1995.
- [6] T. Canfield, T. Disz, M. Papka, R. Stevens, M. Huang, V. Taylor, and J. Chen, "Toward Real-time Interactive Virtual Prototyping of Mechanical Systems: Experiences Coupling Virtual Reality with Finite Element Analysis," In Proceedings of High Performance Computing '96, pp. 339-345, New Orleans, April 1996.
- [7] A. Johnson, M. Roussos, J. Leigh, C. Barnes, C. Vasilakis, and T. Moher, "The NICE Project: Learning Together in a Virtual World," In Proceedings of VRAIS '98, pp. 176-183, Atlanta, March 1998.
- [8] VKH, <http://vkh3.kisti.re.kr/>, 2000.
- [9] NLM, http://www.nlm.nih.gov/research/visible/visible_human.html/, 1998.
- [10] K. Akeley, "RealityEngine Graphics," In Proceedings of ACM SIGGRAPH '93, pp. 109-116, 1993.
- [11] T. J. Cullip and U. Neumann, "Acceleration Volume Reconstruction with 3D Texture Hardware," Technical Report TR93-027, Department of Computer Science, Univ. of North Carolina-Chapel Hill, 1994.
- [12] A. V. Gelder and K. Kim, "Direct Volume Rendering with Shading via Three-Dimensional Textures," In Proceedings of IEEE Volume Visualization '96, pp. 23-30, October 1996.
- [13] F. Dacheil, K. Kreeger, B. Chen, I. Bitter and A. Kaufman, "High-Quality Volume Rendering Using Texture Mapping Hardware," In Proceedings of Eurographics/SIGGRAPH Workshop on Graphics Hardware '98, pp. 69-76, Lisbon, Portugal, October 1998.
- [14] I. Ihm, R. K. Lee, and S. I. Kim, "On Improving Shading for 3D Texture Mapping Based Volume Rendering," manuscript, 2000.
- [15] M. Levoy, "Display of Surfaces from Volume Data," IEEE Computer Graphics and Applications, Vol. 8, No. 5, pp. 29-37, May 1988.
- [16] M. Levoy, "Efficient Ray Tracing of Volume Data," ACM Transactions on Graphics, Vol. 9, No. 3, pp. 245-261, July 1990.
- [17] P. Bourke, Calculating Stereo Pairs, <http://astronomy.swin.edu.au/pbourke/stereographics/stereorender/>, 1999.
- [18] P. Bourke, Generating autostereoscopic lenticular images, <http://astronomy.swin.edu.au/pbourke/stereographics/lenticular/>, 1999.
- [19] EVL, <http://www.evl.uic.edu/>, 2001.
- [20] ReaCTor, <http://www.trimension-inc.com/>, 2001.
- [21] SGI, <http://www.sgi.com/onyx3000/3400.html/>, 2002.
- [22] TrackdAPI, http://www.vrco.com/products/trackd_api.html/, 2001.

임 무 진

2000년 2월 서강대학교 컴퓨터학과 졸업(공학사). 2002년 2월 서강대학교 컴퓨터학과 졸업(공학석사). 현재 엔젠테크놀로지 부설 연구소 재직. 관심분야는 볼륨 렌더링, 3D 애니메이션 등



이 중 연

1998년 2월 서강대학교 컴퓨터학과 졸업(공학사). 2000년 2월 서강대학교 컴퓨터학과 졸업(공학석사). 현재 한국과학기술정보연구원 슈퍼컴퓨팅센터 재직. 관심분야는 과학적 가시화, 가상현실 등



조 민 수

1996년 2월 연세대학교 대기과학과 졸업(이학박사). 현재 한국과학기술정보연구원 슈퍼컴퓨팅센터에 재직(슈퍼컴퓨팅응용실장). 관심분야는 과학적 가시화, VR 가시화, 가상협업환경 구축 등



이 상 산

1992년 4월 Stanford Univ. Mechanical Engineering 졸업(공학박사). 현재 한국과학기술정보연구원 슈퍼컴퓨팅센터 재직(슈퍼컴퓨팅센터장). 관심분야는 Turbulence Simulation, Grid Computing High Performance Networking 등



임 인 성

1985년 2월 서울대학교 계산통계학과 졸업(이학사). 1987년 5월 Rutgers-The State University of New Jersey 컴퓨터학과 졸업(이학석사). 1991년 7월 Purdue University 컴퓨터학과 졸업(이학박사). 1999년 7월 ~ 2000년 6월 University of Texas at Austin의 TICAM(Texas Institute of Computational and Applied Mathematics) 방문 연구원. 1993년 3월 ~ 현재 서강대학교 컴퓨터학과 교수. 관심분야는 컴퓨터 그래픽스, 과학적 가시화, 고성능 계산 등