

다중스크린의 연속적인 배치를 지원하는 몰입형 가상환경 디스플레이 시스템의 개발

남상훈*, 채영호**, 강재훈***

Development of an Immersive VR Display System Supporting Continuous Arrangement of Multiple Screens

Nam, S. H*, Chai, Y. H** and Kang, J. H.***

ABSTRACT

The suggested modular projection system can have flexible arrangements of screens. Modular and continuous arrangements of the display system enable us to modify the screen configuration easily, so that the system can be used in various applications. The image of each screen is calculated automatically by using window projection and the tracked position of the viewer. This system also uses the off-axis stereoscopic projection for the seamless stereoscopic edge blending of multiple connections of screens. The system has been successfully tested for the general navigation model and the CAU driving simulator with motion platform.

Key words : Virtual reality, Immersive display system, Modular projection, Window projection, Off-axis stereoscopic projection

1. 서 론

시뮬레이션을 목적으로 하는 어플리케이션을 개발하는 경우는 시각적인 효과를 사용자에게 보여주는 디스플레이 시스템과 어플리케이션의 용도에 따른 장비들이 하나의 통합 시스템으로서 사용된다. 이 통합 시스템은 사용자에게 실제감을 주기 위하여 시각적 효과, 운동감의 상호 작용, 청각적 효과, 촉각적 효과 그리고 다른 여러 가지 효과들을 사용한다. 이러한 효과들 중 실제감을 주는데 가장 큰 비중을 차지하는 것이 시각적인 효과이다^[1,2].

시각적 효과를 나타내는 디스플레이 시스템은 사용자에게 실제감을 줄 수 있도록 하나의 스크린을 이용한 싱글 스크린으로부터 여러 개의 스크린을 연결하여 놓은 멀티 스크린과 곡면 스크린, 나아가 사용자를 둘

러싸는 6개의 스크린으로 구성되는 C6에 이르기까지 발전되었다^[3]. 보다 넓고 많은 수의 스크린을 사용하면 사용자에게 큰 시야와 몰입감을 제공 할 수 있는 장점이 있지만 디스플레이 시스템의 크기가 거대해 짐에 따라 작은 공간에서의 설치가 불가능하고 설치 비용이 커지며, 또한 설치된 이후에는 이동이 어려운 단점이 있다. 이러한 단점을 해결하기 위하여 ImmersaDesk와 같이 사용 공간을 최소화 하고 이동이 용이하도록 개발된 디스플레이 시스템은 하나의 스크린 사용으로 몰입감이 떨어진다는 문제점이 있으며, SGI™ Reality Center™ Walls와 같은 제품은 다수의 스크린의 결합이 가능하도록 개발되었지만 스크린 배치의 제한성으로 인하여 일반적인 사용에는 부적합하다.

본 연구에서는 위에서 언급된 여러 디스플레이 시스템의 장점을 살릴 수 있도록 개발된 모듈식 디스플레이 시스템을 제안하였다. 이 시스템은 사용 공간을 최소화 하여 이동이 용이하도록 설계되었으며 스크린의 배치가 용이하도록 하드웨어 및 소프트웨어가 개발되었다.

본 연구에서 개발된 모듈식 디스플레이 시스템의 하드웨어는 후방 투영을 이용함으로써 이동이 용이하

*중앙대학교 영상공학과
**정회원, 중앙대학교 영상공학과
***중앙대학교 건축학과
- 논문투고일: 2001. 7. 26
- 심사완료일: 2001. 9. 17

며, 또한 시스템이 설치되는 사용 공간을 줄일 수 있는 장점이 있다. 프로젝터에서 나온 영상이 어플리케이션의 목적에 따라 자유롭게 배치되는 스크린에 비춰져야 하기 때문에 프로젝터와 스크린, 거울의 각도를 자유롭게 변경될 수 있도록 하는 기구학적 설계가 필요하였다. 스크린의 위치가 결정되면 그에 따른 프로젝터 및 거울의 각도와 위치가 필요하므로 각각의 상대 거리와 각도 산출에 필요한 식을 유도하게 되었다. 프로젝터와 스크린, 반사경은 하나의 모듈로 재조합하여 여러 대의 모듈식 디스플레이 시스템으로 배치함으로써 멀티스크린의 장점인 사용자의 시점을 크게 하였으며, 또한 이동과 설치가 용이하게 하였다.

소프트웨어는 단순한 시뮬레이션으로부터 몰입형 가상 환경에 이르기까지 다양한 사용이 가능하도록 설계되었다. 모듈식 시스템을 지원하기 위하여 스크린의 위치와 평면의 방향 벡터를 받아들이고 각각의 스크린에 대한 투영을 모듈화 하여 소프트웨어를 설계함으로써 스크린의 수에 상관없이 사용할 수 있도록 하였다. 시스템은 윈도우 투영을 구현하여 사용자의 위치에 따라 영상을 정확하게 변화시킴으로써 사용자가 몰입감을 느낄 수 있도록 하였다. 여러 개의 모듈화 된 스크린을 배치하였을 때 스테레오 영상의 연속성을 위해 일반적인 on-axis 투영 방식을 사용하지 않고 off-axis 투영 방식을 사용하였으며 view-volume을 주위의 스크린과 비교하여 변화시킴으로써 스크린들이 만나는 부분에서의 정확한 영상을 얻을 수 있도록 하였다.

이러한 모듈식 디스플레이 시스템의 사용은 어플리케이션의 목적에 따라 다양하게 사용할 수 있다는 장점이 있으며 사용 장소에 따라 이동 및 재배치가 가능하고 소프트웨어로써 해당 투영을 자동으로 계산하기 때문에 적은 시간으로도 해당 어플리케이션을 완성할 수 있다.

2. 모듈식 시스템의 구성

2.1 스크린과 반사경과의 관계

스크린과 프로젝터와의 위치는 프로젝터로부터 나오는 영상이 스크린까지 가는 수평 거리와 수직 거리를 계산함으로써 결정된다. Fig. 1(a)에서와 같이 스크린과 프로젝터만으로 구성되어 있는 일반적인 투영 시스템과는 달리 Fig. 1(b)는 프로젝터와 스크린 사이에 여러 개의 반사경을 위치시킴으로써 사용 공간을 최소화 할 수 있다.

Fig. 2는 하나의 반사경을 사용한 시스템에서 프로젝터의 영상이 나오는 점(C)과 반사경의 윗면(E), 아래면

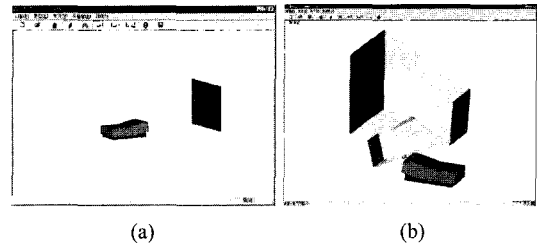


Fig. 1. Simulation of the system configurations.

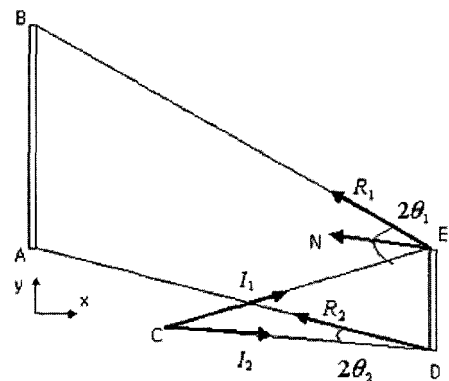


Fig. 2. Relative position vector between the screen and the mirror.

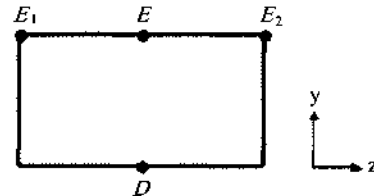


Fig. 3. Y-Z coordinates of the mirror.

(D), 스크린의 윗면(B), 아래면(A)을 x-y평면상에 간략히 나타낸 것이다. Fig. 3은 반사경을 나타내는 (D~E) 구간을 y-z평면상에 나타낸 것이다. 이를 바탕으로 입사 광선의 방향을 나타내는 단위 벡터 I , 반사광선을 나타내는 단위 벡터 R , 입사와 반사가 일어나는 평면에서의 단위 법선 벡터인 N 을 구하면 다음과 같다.

$$N = \frac{\overrightarrow{DE_1} \times \overrightarrow{DE_2}}{|\overrightarrow{DE_1} \times \overrightarrow{DE_2}|} \quad (1)$$

$$\cos \theta_1 = I_1 \cdot N \quad (C-E-B) \text{ 구간}$$

$$\cos \theta_2 = I_2 \cdot N \quad (C-D-A) \text{ 구간}$$

$$I_1 = \frac{\overrightarrow{CE}}{|\overrightarrow{CE}|} \quad I_2 = \frac{\overrightarrow{CD}}{|\overrightarrow{CD}|} \quad (2)$$

위 식을 이용하여 반사각을 나타내는 R_1, R_2 를 구할 수 있다.

(C-E-B) 구간

$$\begin{aligned} \vec{R}_1 &= \vec{I}_1 + 2\vec{N}\cos\theta_1 \\ &= \frac{\vec{CE}}{|\vec{CE}|} + 2 \cdot \frac{\vec{DE}_1 \times \vec{DE}_2}{|\vec{DE}_1 \times \vec{DE}_2|} \left(\frac{\vec{CE}}{|\vec{CE}|} \cdot \frac{\vec{DE}_1 \times \vec{DE}_2}{|\vec{DE}_1 \times \vec{DE}_2|} \right) \end{aligned} \quad (3)$$

(C-D-A)구간

$$\begin{aligned} \vec{R}_2 &= \vec{I}_2 + 2\vec{N}\cos\theta_2 \\ &= \frac{\vec{CD}}{|\vec{CD}|} + 2 \cdot \frac{\vec{DE}_1 \times \vec{DE}_2}{|\vec{DE}_1 \times \vec{DE}_2|} \left(\frac{\vec{CD}}{|\vec{CD}|} \cdot \frac{\vec{DE}_1 \times \vec{DE}_2}{|\vec{DE}_1 \times \vec{DE}_2|} \right) \end{aligned} \quad (4)$$

반사경의 유무와 상관없이 프로젝터의 중심(C)에서 스크린까지의 거리는 항상 일정하다. 이를 이용하면 스크린의 위치와 반사경의 위치가 서로 일정한 관계에 있으므로 스크린의 위치가 정해짐에 따라 반사경의 위치를 구할 수 있다.

$$\begin{aligned} |\vec{CE}| + |\vec{EB}| &= \text{const} & |\vec{CD}| + |\vec{DA}| &= \text{const} \\ P_1 + Q_1 &= \text{const}_1 & P_2 + Q_2 &= \text{const}_2 \\ (P_1 = |\vec{CE}|, Q_1 = |\vec{EB}|, P_2 = |\vec{CD}|, Q_2 = |\vec{DA}|) \end{aligned}$$

$$\begin{aligned} B_x &= E_x + (\text{const}_1 - P_1)R_{1x} \\ B_y &= E_y + (\text{const}_1 - P_1)R_{1y} \\ B_z &= E_z + (\text{const}_1 - P_1)R_{1z} \end{aligned} \quad (5)$$

$$\begin{aligned} A_x &= D_x + (\text{const}_2 - P_2)R_{2x} \\ A_y &= D_y + (\text{const}_2 - P_2)R_{2y} \\ A_z &= D_z + (\text{const}_2 - P_2)R_{2z} \end{aligned} \quad (6)$$

스크린의 위치가 고정되어 있다면 반사경의 위치는

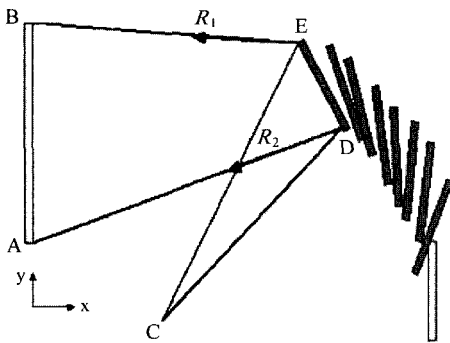


Fig. 4. Simulation of the relative mirror position.

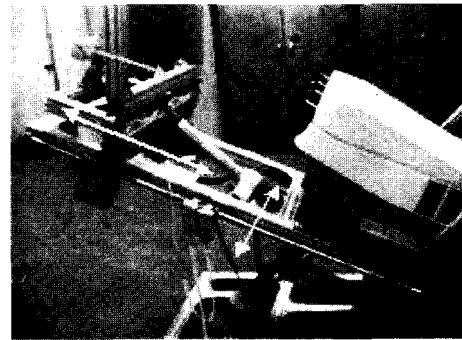
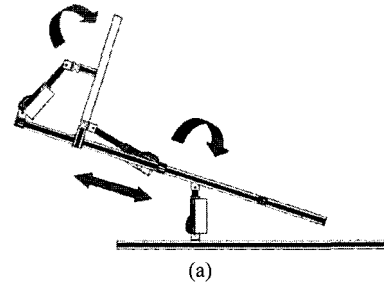


Fig. 5. Actuation connection.

Fig. 4와 같이 수치 대입법과 시뮬레이션을 통하여 알아낼 수 있다.

2.2 시스템의 하드웨어 구성과 설계

본 모듈식 시스템은 프로젝터, 스크린, 반사경으로 이루어져 있으며 프로젝터는 BARCO-GRAPHICS 808s, 스크린은 Stewart Filmscreen사의 OptaWave, 반사경은 탐비전 코리아 제품, 그리고 구동부는 Dewar사의 DYMATE Actuator를 사용하였다.

또한, 이동과 설치가 쉽고 빠르도록 프레임의 재질을 알루미늄으로 하였고 각 부분은 나사로 연결되었다. 각 부분별 움직임은 Fig. 5와 같이 프로젝터와 반사경 부분이 동시에 직진 및 회전 운동을 하며, 다시 반사경은 독립적으로 직진, 회전 운동을 할 수 있게 설계하였다. 이러한 움직임은 선형 Actuator를 통하여 전동식으로 이루어진다.

3. 모듈식 시스템의 소프트웨어 설계

모듈식 시스템의 소프트웨어 부분은 일반적인 디스플레이 시스템뿐만 아니라 몰입형 가상 환경을 지원할 수 있도록 구성하였다. 또한, 여러 개의 스크린 사용을 기본으로 지원하고 스테레오 영상을 스크린의 수에 관

계없이 연속성을 갖도록 하였다.

3.1 윈도우 투영

일반적인 투영 방식은 카메라 투영 방식으로서 사용자의 눈의 위치와 투영 평면(스크린)의 위치가 일정한 것으로 가정한다. 카메라 투영은 Fig. 6과 같이 사용자의 움직임에 따른 위치가 영상에 영향을 줄 수 없고 단지 입력 장치를 이용하여 전체적인 투영 평면을 회전하거나 이동함으로써 사용자가 움직이는 것과 같은 효과를 준다. 그러나 윈도우 투영은 Fig. 7과 같이 투영평면과 사용자의 위치가 자유롭기 때문에 투영 평면의 회전과 이동, 그리고 공간 추적 장치를 이용한 사용자의 눈의 위치를 알아냄으로써 영상에 반영하게 된다. 그러므로 사용자가 스크린의 전방 또는 후방 그리고 좌우로 움직임에 따라 영상이 변하기 때문에 몰입감을 느낄 수 있게 된다.

윈도우 투영의 가장 중요한 요소는 시점과 투영 평면이 자유로워야 한다는 점이다. 이와 같은 윈도우 투영의 구현을 위해 OpenGL API¹⁶⁾에서 제공하는 투영

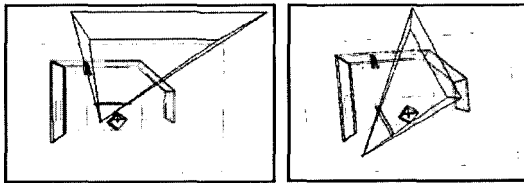


Fig. 6. Camera projection.

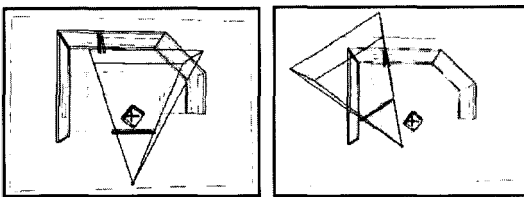


Fig. 7. Window projection.

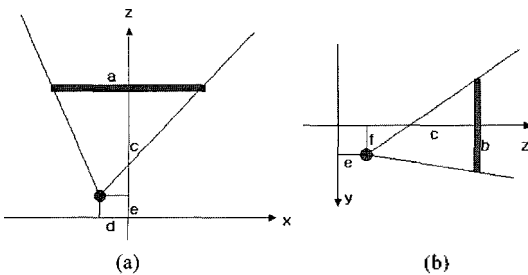


Fig. 8. Projection parameters for voluntary viewpoint.

함수인 $glFrustum()$ 을 이용하여 시점에 따른 스크린의 clipping volume값을 조정하게 되었다. 이 시스템에서는 $scr_win(left, right, bottom, top)$ 의 형태의 함수를 정의하여 left, right, bottom, top clipping volume값을 매개 변수로 받아 사용하였다.

윈도우 투영에서의 시점은 사용자가 움직임에 따라 공간 추적 장치에서 얻어진다. 이 시점을 투영에 사용하게 되며, 투영 함수의 매개 변수는 움직이는 시점과 스크린의 위치에 따른 비례식으로 구성된다. Fig. 8은 스크린의 좌표계 중심에서 z축으로 c만큼 떨어져 있고 시점은 스크린 좌표계의 중심에서 d, e, f만큼 이동되어 있는 경우로서 투영 함수의 매개 변수 값은 다음과 같다.

$$scr_win\left(\frac{(a+2d)}{2(c-e)}, \frac{(a-2d)}{2(c-e)}, \frac{(b-2f)}{2(c-e)}, \frac{(b+2f)}{2(c-e)}\right) \quad (6)$$

또한, 모듈식 프로젝션 시스템에서는 스크린의 배치가 자유로워야 하므로 Fig. 9와 같이 스크린의 중심이 스크린의 좌표계의 중심에서 g, h의 크기 만큼 이동하여 위치하여야 한다. 시점이 좌표계의 중심에서 d, e, f의 크기 만큼 이동되어 있을 때의 투영 함수의 매개 변수 값을 구해보면 다음과 같다.

$$scr_win\left(\frac{2(g-d)-a}{2(c-e)}, \frac{2(g-d)+a}{2(c-e)}, \frac{2(h-f)+b}{2(c-e)}, \frac{2(h-f)-b}{2(c-e)}\right) \quad (7)$$

3.2 다중 스크린의 배치

여러 개의 스크린을 이용하는 어플리케이션은 사용자에게 더 넓은 시야를 제공함으로써 보다 나은 실재감을 줄 수 있다는 장점이 있다¹⁵⁾. 이러한 스크린의 수와 위치는 어플리케이션의 용도에 따라 정해진다. 스크린의 수와 위치를 자유롭게 배치할 수 있으면 여러

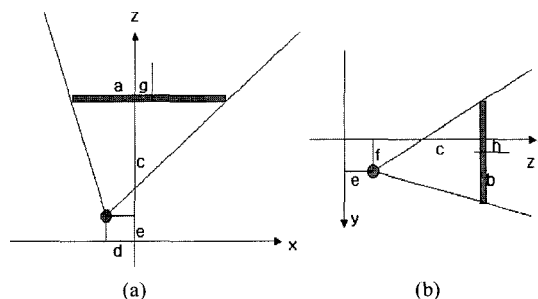


Fig. 9. Projection parameters for different screen configuration (a) Top view (b) Side view.

어플리케이션에 다양하게 사용할 수 있다는 장점이 있다⁶⁾. 스크린은 모듈식으로 각각의 지역 좌표계를 가지고 있도록 설계되었다. 그 좌표계들은 하나의 시스템에서 동기화 될 수 있도록 Fig. 10의 표시와 같이 Frame_Center(Fc)의 기초 좌표계를 가지고 있으며 그 위에 각각의 스크린에 대한 지역 좌표계가 기초 좌표계를 중심으로 놓여진다. Fig. 10과 같이 두개의 스크린으로 구성된 시스템을 예로 들면 전체 시스템은 Fc 라는 기초 좌표계 위에 Screen1(S1)의 지역 좌표계인 F1과 Screen2(S2)의 지역 좌표계인 F2를 위치시키게 된다. 시스템은 기초 좌표계에 배치된 지역 좌표계에 대한 정보와 기초 좌표계에서의 P점의 좌표를 가지고 있다. 스크린에 영상을 보여주기 위하여 Fig. 11와 같이 기초 좌표계에서의 P점을 스크린 지역 좌표계에서의 P점 좌표로 변환한 후 각각의 스크린에 대해 지역 좌표계를 기준으로 윈도우 투영을 하게 된다.

3.3 Off-axis투영방식

스테레오 영상을 구현하는 방법은 크게 on-axis투영 방식과 off-axis투영 방식으로 나눌 수 있다⁷⁾. 여기에서 on-axis투영 방식은 계산량이 적어 간편하므로 가장 널리 사용된다. 또한, 카메라 투영 방식으로서 스테레

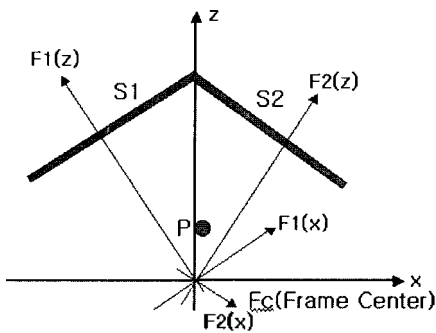


Fig. 10. Coordinates for dual screen system.

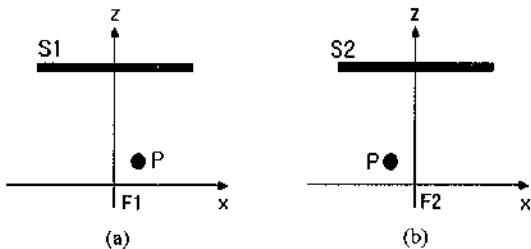


Fig. 11. Viewpoints in local coordinate system for each screen.

오 영상을 구현할 수 있기 때문에 이전의 윈도우 투영 방식을 지원하지 않는 그래픽 라이브러리에서 사용되었다. On-axis투영 방식은 왼쪽 눈의 좌표를 오른쪽 눈의 위치에 대하여 일정한 거리 만큼 x축으로 이동시켜 사용하는 방식이다. 하나의 스크린을 기준으로 사용하는 경우는 문제가 발생하지 않으나 여러 개의 스크린이 연결된 경우는 정확한 스테레오 이미지를 표시하지 못한다. Fig. 12에서와 같이 두 개의 스크린이 연결되어 있을 경우 on-axis투영 방식을 사용하면 기준이 되는 왼쪽 눈은 L로서 같은 좌표를 갖는다. 그러나 오른쪽 눈의 위치는 스크린1(S1)의 지역 좌표계인 F1에 대하여 R1, 스크린2(S2)의 지역 좌표계인 F2에 대하여 R2라는 서로 다른 위치를 가지게 되므로 Fig. 13에서와 같이 왼쪽 눈의 영상은 스크린의 연결 부분에서 이미지의 연속성이 있으나 오른쪽 눈의 영상은 이미지의 불연속성이 나타나게 된다.

Off-axis투영 방식을 사용하면 Fig. 14와 같이 스크린이 연결되어 있는 경우 왼쪽 눈과 오른쪽 눈의 좌표

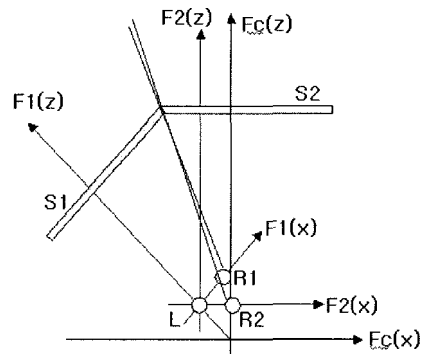


Fig. 12. The discontinuity of the on-axis perspective projection in dual screen configuration.

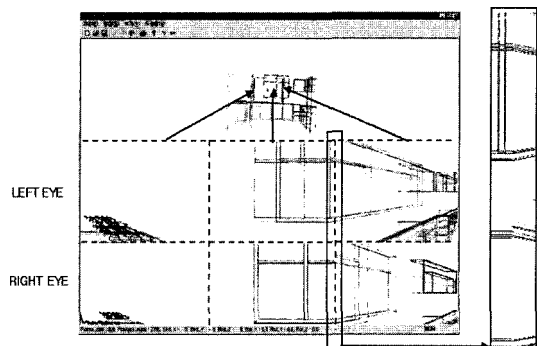


Fig. 13. The discontinuous image of the on-axis perspective projection.

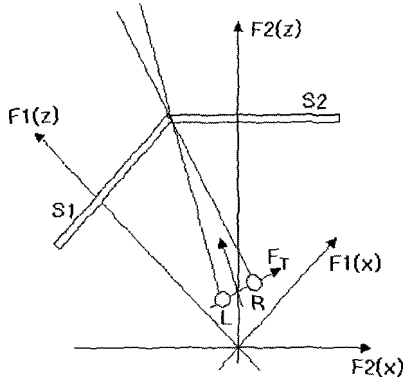


Fig. 14. The continuity of the off-axis perspective projection in dual screen configuration.

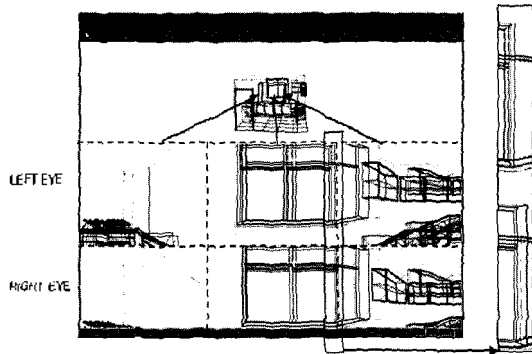


Fig. 15. The continuous image of the off-axis perspective projection.

는 스크린1(S1)의 좌표계(F1)와 스크린2(S2)의 좌표계(F2)와 관계없이 눈의 위치를 알아내기 위한 공간 추적 장치의 지역 좌표계를 나타내는 \$F_T\$에 의하여 구해진다. \$F_T\$의 왼쪽 눈(L)과 오른쪽 눈(R)은 좌표계 변환에 의하여 각각의 스크린 지역 좌표계에 맞게 변환되어 사용되므로 항상 일치한다. 따라서 전체적인 스크린 상 의 눈의 위치는 동일하게 되므로 스크린과 스크린 사이의 영상이 연속성을 가지게 된다. 이 Off-axis투영 방식은 윈도우 투영이 선행될 때 구현되므로 윈도우 투영을 지원하지 않는 프로그램에서는 사용할 수 없다.

3.4 View Volume의 조정

투영에 의한 영상을 만들어 내는 경우 view volume이라는 잘린 사각뿔 공간 안에 존재하는 오브젝트들만 투영하여 영상을 만들어 낸다. 적당한 view volume을 조절함으로써 투영에서 보이지 않는 부분에 대한 폴리곤의 수를 감소시켜 렌더링 속도를 증가시킬 수 있다.

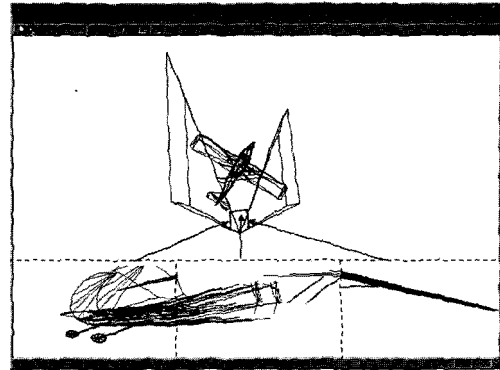


Fig. 16. Unbalanced far clipping volume.

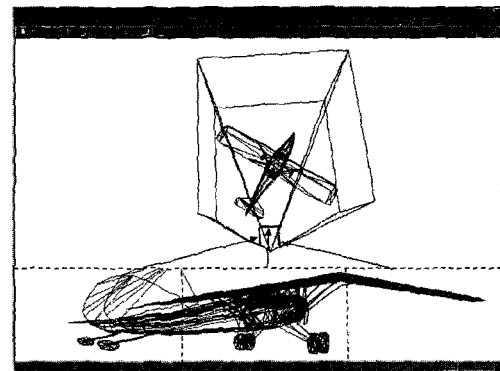


Fig. 17. Effective far clipping volume.

View volume을 조절해 주기 위하여 near clipping값과 far clipping값을 지정해 줌으로써 view volume을 정하게 된다. 그러나 다수의 스크린을 가지는 시스템에서 윈도우 투영을 사용할 경우는 각각의 시점과 스크린과의 거리가 서로 다르게 된다.

이에 따라 시점이 변할 경우 스크린 배개 변수 값들이 변하기 때문에 각각의 스크린에 대한 지역 좌표계에서 일정한 near, far clipping값이 존재하여도 실제적인 near, far clipping volume의 위치가 변하게 된다. Near clipping값은 크게 변하지 않아 영상에 영향을 미치지 않으나 far clipping값의 차이는 스크린간의 이미지의 불일치가 일어나게 한다.

예를 들면, Fig. 16과 같이 다수의 스크린이 배치되어진 경우 스크린이 겹쳐지는 부분에서 서로 다른 far volume의 위치를 가지게 되고 이미지의 불연속성이 나타날 수 있다. 연결된 스크린의 경우는 실제적인 far volume의 값을 각각의 스크린 지역 좌표계에 대하여 윈도우 투영이 변할 때마다 전체 시스템에서의 far volume값을 계산하여 교체준다. 이는 Fig. 17에서와 같

이 view volume의 값이 주위의 스크린과 비교하여 변함으로써 투영된 영상 부분이 동시에 보이거나 보이지 않게 되어 스크린 사이의 연속된 이미지를 제공해 주어야 한다.

4. 모듈식 시스템의 구축 예

4.1 시스템의 구성

모듈식 시스템의 활용과 실험을 위하여 직접 제작한 모듈식 디스플레이 하드웨어를 대신하여 두 대의 모니터와 두 대의 컴퓨터를 모듈식 시스템으로 구성하여 실험을 하였다. 본 시스템은 Device-based 소프트웨어 모델을 기초로 하여 설계되어 졌으며, Fig. 18과 같이 각각의 모듈식 시스템을 초기화하고 서버 컴퓨터에서 공간 추적 장치 및 스페이스볼의 디바이스 데이터를 입력 받도록 되어 있다. 입력된 디바이스 데이터를 이용하여 두 눈의 위치를 계산하고 이를 클라이언트 컴퓨터로 보내주게 된다. 사용자는 스테레오 안경을 쓰고 몰입된 가상 환경을 느낄 수 있다. 각각의 컴퓨터는 데이터를 주고 받으며 시스템을 동기화 한다. 이 시스템의 소프트웨어는 크게 두 부분으로 나누어져 있는데 전체적인 시스템을 구성하기 전에 가상 환경을 구성하거나 시험하는 시뮬레이션 프로그램과 실제로 몰입된 가상환경을 느끼도록 지원해주는 네비게이션 프로그램으로 이루어진다.

4.2 시뮬레이션 프로그램

시뮬레이션 프로그램에서는 어플리케이션에 사용될 어질 가상 환경을 구성해 볼 수 있으며 스크린의 수와 위치 데이터를 받아 배치시킨 후 스크린의 위치를 변경시키면서 시뮬레이션 할 수 있다.

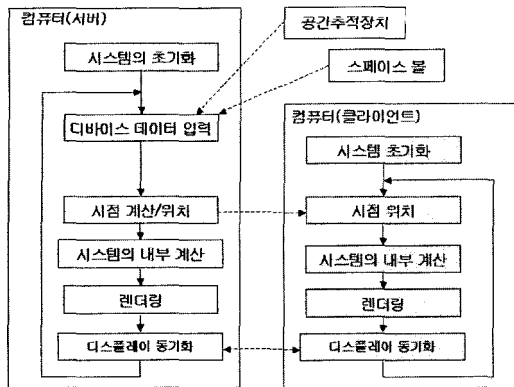


Fig. 18. Data flow of the display system.

가상 환경을 구축하는 방법으로는 각각의 폴리곤을 사용하거나 여러 모델링 프로그램을 이용하여 만들어진 데이터를 이용할 수 있다. 본 프로그램에서는 3D Studio MAX에서 얻을 수 있는 '3ds' 파일과 CAD 데이터인 'dxf' 파일, 그리고 인터넷에서 가상 환경

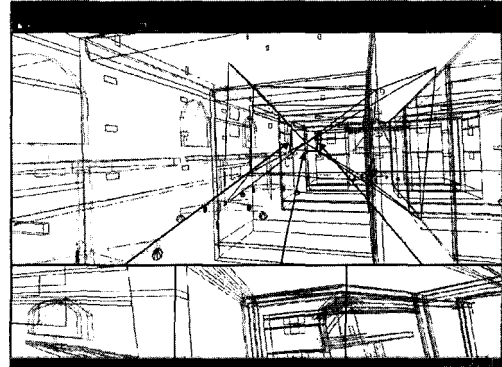
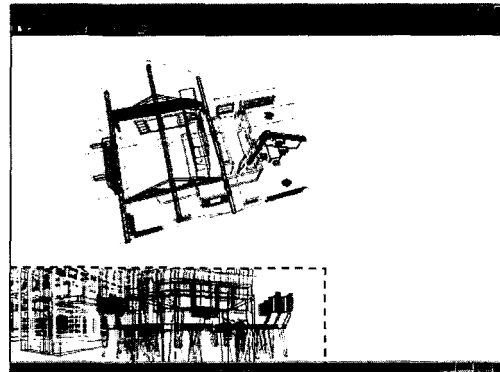
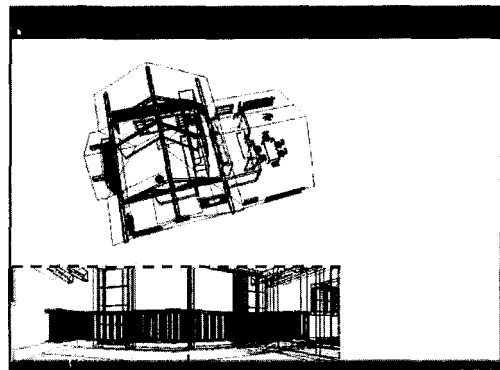


Fig. 19. Example of the simulation program.



(a)



(b)

Fig. 20. Different projected images by changing the screen locations.

구현에 많이 사용되는 'wrl' VRML파일을 사용하도록 하였다. 이를 이용하여 각각의 파일을 불러들여 scenegraph의 데이터 구조로 배치시킨 후 VRML방식으로 저장하여 사용하게 된다. 다른 형식의 파일은 폴리곤의 형태로 저장되어 데이터를 수정하기 어려우나 VRML파일 형태는 트리 구조를 유지하고 있어 데이터를 수정할 수 있다.

또한, Fig. 19와 같이 어플리케이션에 사용될 스크린의 수와 위치를 배치시켜 놓은 후 프로그램 상단에서 배치된 스크린의 위치를 확인, 수정할 수 있으며 프로그램 하단에서 각각의 스크린에 나타날 영상을 확인해 볼 수 있다. 또한, Fig. 20에서와 같이 스페이스볼을 이용하여 스크린을 움직이면서 네비게이션을 할 때 나타나는 영상을 시뮬레이션 할 수 있다.

4.3 네비게이션 프로그램

네비게이션 프로그램은 실제 가상 환경을 체험하기 위해서 사용되는 영상이다. Fig. 21에서와 같이 시뮬레이션 프로그램의 하단부에 보여졌던 각각의 영상이 Fig. 22에서와 같이 하나의 스크린에서 전체 화면 크기로 나타나며, 이는 서버와 클라이언트간의 TCP/IP 통신을 통해서 구현된다.

서버와 클라이언트 컴퓨터간의 데이터 통신은 Fig. 23에서와 같이 이루어지며 서버에 연결되어 있는 공간

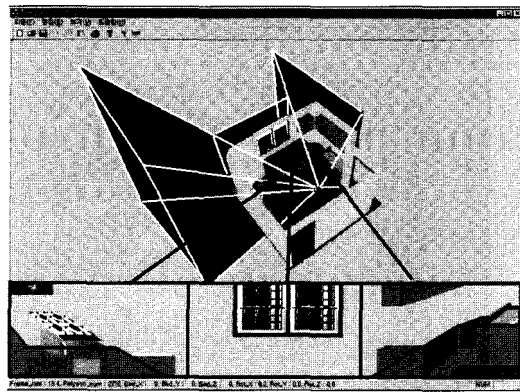


Fig. 21. Example of the simulation program.

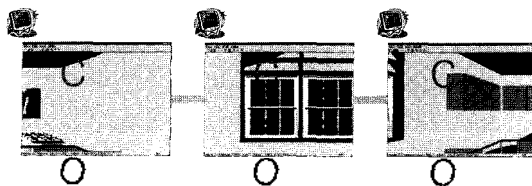


Fig. 22. The image of each monitor(screen).

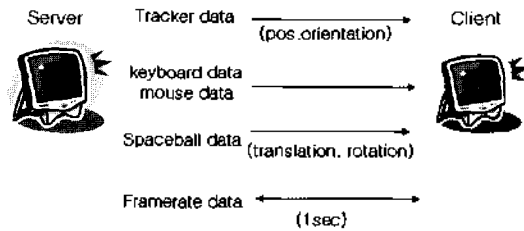


Fig. 23. Data communication between the server and the client.

추적 장치로부터 입력 받은 데이터를 실제 예상되는 눈의 위치 데이터로 변환시킨 값과 스페이스 볼에서 얻어지는 데이터를 클라이언트 컴퓨터로 보내주게 된다. 또한, 키보드와 마우스에 의한 데이터도 보내준다. 각각의 컴퓨터를 통해 영상이 보여질 경우 각각의 영상에서 표시되는 물체들의 폴리곤의 수나 텍스처 매핑 등의 특징에 따라 컴퓨터의 디스플레이 부분에서의 계산량은 많은 차이를 보이게 된다. 그러므로 framerate가 변하게 된다. 이때 동일한 framerate를 유지하기 위하여 컴퓨터 사이에서 framerate를 주고 받게 되며 이 중에서 낮은 framerate로 동기화시키게 된다.

5. 결 론

본 연구에서는 모듈식 디스플레이 시스템의 개발을 위한 하드웨어 구성 요소와 다중 투영 소프트웨어를 개발하였다.

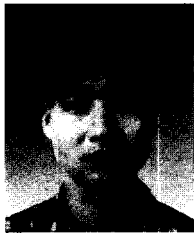
시스템의 크기를 줄이고 좁은 공간에서의 설치를 가능하게 하기 위해 프로젝터와 스크린 사이에 반사경을 설치하였으며, 또한 스크린의 위치를 자유롭게 배치할 수 있도록 프로젝터 및 반사경의 각도와 위치가 변경될 수 있는 기구학적 설계를 하였다. 모듈식 시스템을 사용하여 자유롭게 스크린을 변화함으로써 여러 가지 시뮬레이션 분야에 다양한 디스플레이 시스템으로 사용할 수 있도록 하였다. 다수의 스크린이 자유로운 위치에 배치되는 모듈식 시스템을 위한 다중 투영 소프트웨어를 개발함으로써 스크린의 수에 상관없이 쉽게 어플리케이션에 적용할 수 있다. 이 시스템은 몰입형 가상 환경을 위하여 공간 추적 장치와 스페이스볼을 입력 장치로 사용할 수 있으며, 또한 스테레오 안경을 통하여 단순한 가상 환경의 네비게이션은 물론 몰입형 어플리케이션에도 사용할 수 있다. 나아가 서버 컴퓨터와 클라이언트 컴퓨터를 TCP/IP통신으로써 동기화하여 개인용 컴퓨터의 연결을 이용한 대형 가상 공간의 구축이 가능하도록 하였다.

감사의 글

본 연구는 과학기술부 국가지정연구실사업(2000-NL-01-C-285)과 1999년도 중앙대학교 학술연구비 지원으로 수행되었음.

참고문헌

1. John Vince, *Virtual Reality Systems*, Addison-Wesley, pp. 258-281, 1995.
2. Grigore Burdea, Philippe Coiffet, *Virtual Reality Technology*, John Wiley & Sons, New York, pp. 221-255, 1994.
3. Mary Cole, James Gruening, "Choosing a Screen Configuration to Suit the Application," *The Fourth International Immersive Projection Technology Workshop* 2000, 2000.
4. Neider, J. Davis, T. and Woo, M., *OpenGL Programming Guide*, Addison-Wesley, 1993.
5. Mark Meyer, Alan H. Barr, "ALCOVE," *Design and Implementation of an Object-Centric Virtual Environment*, *IEEE Virtual Reality*, pp. 46-52, 1999.
6. Nam, S. H, Lee D. H, Im, J. H, and Chai, Y. H, "Dual Projection-based VR System for the light weight Motion-based Driving Simulator," *VRST 2000*, pp. 195-198, 2000.
7. Larry F. Hodges, "Basic principles of stereographic software development," *Stereoscopic Displays and Applications*, SPIE Vol. 1457, 1991.
8. C. Cruz-Neira, "Virtual Reality based on multiple projection screens : The CAVE and its applications to computational science and engineering," Ph.D. Dissertation, University of Illinois at Chicago, May 1995.



남 상 훈

1999년 중앙대학교 기계공학과 학사
2001년 중앙대학교 첨단영상대학원 영상공학과 석사
2001년~현재 현대·기아 연구개발본부 연구원
관심분야 : Virtual Environments



채 영 호

1989년 중앙대학교 기계공학과 학사
1989년~1992년 삼성전기 CAD/CAM실 연구원
1994년 SUNY at Buffalo 기계공학과 석사
1997년 Iowa State University 기계공학과 박사
1998년 CASE Virtual Prototyping Lab Consultant
1998년~1999년 중앙대학교 기계공학부 조교수
1999년~현재 중앙대학교 첨단영상대학원 조교수
관심분야: Haptics, Virtual Environments



강 재 훈

1991년 중앙대학교 건축학과 학사
1994년 The Ohio State University, 구조공학과 공학석사
1997년 The Ohio State University, 구조공학과 공학박사
1998년 경주대학교 전임강사
2000년~현재 중앙대학교 건축학과 조교수
관심분야 : Dynamic System