

□신기술해설□

가상 스튜디오

최영진[†] 남승진^{††} 최창락^{††}

◆ 목 차 ◆

- | | |
|-----------------|---------------|
| 1. 서론 | 4. 카메라 캘리브레이션 |
| 2. 발전방향 | 5. 소프트웨어 개발 |
| 3. DreamSet 시스템 | 6. 결론 |

1. 서론

뉴스, 일기 예보, 대담 프로 등에서 크로마키어 기술이 자주 사용된다. 이 기술은 배우와 실제 소품들을 블루 박스 앞에서 촬영한 후에 이 영상을 기 제작된 배경 영상과 합성하는 것이다. 이러한 방식은 카메라의 움직임에 따라 배경 영상이 연동되지 않기 때문에 부자연스럽고 사용상 제약이 많이 따른다. 이에 대한 개선 방안으로 가상 스튜디오 기술이 제시되었으며, 이 방식은 카메라 움직임에 연동하여 배경을 자동으로 변화시킨다. 이렇게 함으로써 전경과 배경의 시점을 일치시킬 수 있다. 이것을 가장 자연스럽게 구현할 수 있는 방법은 컴퓨터를 사용하여, 3D 화면(가상 세트)에 대한 2D 원근 영상을 실시간으로 생성하는 것이다. 2D 원근 영상은 카메라 헤드로부터 입력되는 정보에 의해서 생성된다.

결과적으로 배우를 2D 원근 영상과 합성함으로써, 실제 3D 화면에서 배우가 움직이는 듯한 효과를 얻을 수 있다. 가상 스튜디오 기술은 전경과 배경의 시점을 일치시키는 것 외에도 배우가 가상 물체에 의해 전체 또는 일부가 가려지는 효과가 가능하다. 이 기술은 학문적으로 많이 연구되어 왔었지만, 컴퓨터의 속도가 실시간 처리를 지원하지 않음으로 해서 전용 하드웨어를 제작하거나 포스트 프리덕션 방식으로 사용되어 왔다. 최근 가상 현실, 3차원 시뮬레이션 등과 같은 분야에서 실시간 3차원 그래픽의 수요가 늘어감에 따라 3차원 그래픽의 속도를 대폭 향상시킨 컴퓨터의 개발이 이루어졌다.

가상 스튜디오는 블루 박스보다 훨씬 큰 세트를 생성할 수 있으며, 마치 환상과도 같은 거대한 세트도 만들 수 있다. 물리적으로 구성이 어렵거나 너무 비싸서 설치가 불가능한 세트를 구성하는 것도 가능하다. 우주공간에서 거대한 지구가 회전하는 것과 같은 특수 효과도 가능하게 되었다. 세트 수정은 쉽고 효과적이며, 설치, 해체, 보관에서 발생하는 문제가 없으므로, 스튜디오의

† 정회원 : KBS 기술연구소 차장

†† 정회원 : KBS 기술연구소 연구원

사용 효율을 극대화할 수 있다. 현재는 컴퓨터의 처리 속도가 부족하여 배경 화면의 사실감이 떨어지며, 배경화면과 연기자의 조화가 이루어지지 않아 어색한 장면이 연출되기도 한다.

그러면 실제 세트 제작보다 훨씬 많은 제작비와 시간이 투입되며 아직은 구현된 세트의 현실성이 부족한 가상 스튜디오에 왜 그렇게 많은 방송관계자들이 열광적인 관심을 나타내는가? 그동안 한정된 몇몇 채널로 유지되던 TV 방송이 지역 민간 TV 방송, 케이블 TV 방송의 등장으로 수십 개의 채널로 증가하였으며, 위성 TV 방송 등 각종 새로운 매체의 허가가 계속적으로 이어지고 있다. 이와 대조적으로 광고 시장은 정체되어 있거나 그 증가 속도가 매우 완만하다. 새로이 허가된 방송 채널들이 활성화되는 몇 년 후에는 지금보다 더욱 저렴한 제작비로 더욱 우수한 프로그램의 제작이 요구되며 이러한 요구에 대한 해결책의 하나로 가상 스튜디오가 주목 받고 있다.

가상 스튜디오의 특징으로는 실시간으로 배우와 3D 배경화면을 합성할 수 있을 뿐만 아니라 동시에 문자, 그래프, 도형, 입체 모형 등 다채로운 표현이 가능하다. 데이터만 미리 만들어져 있으면 세트 내에서 자유롭게 연출이 가능하며 자유롭게 시점을 제어할 수 있어 갑작스런 상황의 변화에도 대처할 수 있어, 속도 방송이나 전화 앙케이트 등에 이용이 가능하다. 또한 현실로 제작이 불가능한 세트나 촬영이 불가능한 장소 등을 가상으로 제작하여 보여줄 수 있다.

2. 발전방향

그 동안 RT-SET, Elset, FOR.A, Digimedia, Discreet Logic사 등의 선발사 제품들은 SGI의 Onyx 워크스테이션을 사용한 고가의 시스템을 구성하여 방송 관계자들의 관심을 끌었으며, 가상

스튜디오의 가능성과 발전성을 위주로 시장 개척의 의도가 강하였다. '97년에는 Evans+Sutherland사의 mindset 등 Window NT를 사용하여 시스템을 제어 및 조정하고 그래픽 처리는 전용 엔진을 사용하는 시스템이 다수 발표되었다. 또한 SGI의 워크스테이션 중에서 비교적 저가인 O2 및 Impact 등을 사용하여 시스템의 가격을 대폭 낮추어 소형 프로그램 제작자들도 쉽게 구입할 수 있도록 하여 시장의 확대를 목표로 하고 있다. 시스템의 방향은 Onyx IR 계열과 Window NT 계열로 나뉘었으며, On-Air가 가능한 고가 시스템과 녹화 및 후처리 등으로 보완되는 저가 시스템으로 나뉘어져 있다. 이러한 시스템은 기존 고가 시스템에 비하여 가격이 대폭 낮아진 반면 기능면에서는 아직 부족하였으나 하드웨어의 빠른 발전 추세를 감안하면 2 - 3 년이면 만족할 만한 기능을 갖출 것으로 생각된다.

<표 1> 가상 스튜디오의 발전 추세

1.	'95년	제품 발표기	<ul style="list-style-type: none"> • 초기 제품 발표 단계 • 사용자 환경이 부족하며 시스템 불안정
2.	'96년	제품 안정기	<ul style="list-style-type: none"> • 대부분 Onyx 워크스테이션 사용 • 안정된 시스템 출시
3.	'97년	시장 개척기	<ul style="list-style-type: none"> • 저가격의 NT 계열 시스템 발표 • 가상 텔런트 등 다양한 기능 부가 • S/W 가격 하락 (\$60만 → \$20만)
4.	'98년	시장 확대기	<ul style="list-style-type: none"> • 가격과 성능 만족하는 시스템 출현 예상 • 시스템 보급 확대

상용 벤더의 제품 동향을 보면 RT-SET사는 기존의 Larus 및 Otus외에도 O2 및 Impact 등을 포함하여 SGI의 워크스테이션에 사용할 수 있는 새로운 3D 가상 스튜디오 시스템을 선보였다. 새로이 업그레이드된 기능으로는 그림자 기능, Prompter (Actor에게 이동 방향 및 위치 등을 보여줌)

기능, Indigo Impact를 사용한 Preview 기능, 입의 CG(문자발생기) S/W 혹은 Motion Capture 시스템 등의 사용으로 충실한 시스템의 구현을 목표로 하고 있다. Digimedia사의 Brainstorm은 가장 먼저 Onyx2를 사용한 시스템을 발표하였다. 센서 및 위치 이동을 위한 보조 장치 등을 통한 여러 가지의 표현 기법의 다양화 등 프로그램의 실질적인 제작에 중점을 두고 있다. FOR.A 및 Orad의 시스템은 패턴 인식 방법을 사용한 견착식 혹은 이동 카메라를 사용하는 독특한 방법으로 장단점을 갖고 있다. 동사 Cyberset E는 SGI O2 워크스테이션을 사용하는 저가형의 가상 세트 시스템이다. 가상 인물 performance animator, actor feedback 시스템의 채택이 특징이다. Discreet Logic사에서는 가상 스튜디오 시스템인 Vapour의 버전 2.0을 발표하였다. 상당한 수준의 3차원 모델링 toolset를 제공함으로써 제작자가 3D 그래픽들을 즉시 만들 수 있어서 요구되는 제작 스케줄에 맞출수 있는 편리함이 있다. 또한 터치 스크린 등의 외부 기기의 신호와 투표, 증권 시세, 뉴스 데이터 베이스 등의 데이터에 의하여 3D 이미지들과 텍스트를 실시간 렌더링 하는 기능도 갖추고 있다.

가상 스튜디오는 '92년 일본의 참의원 선거에서 후지TV에서 사용하였고, '96년 국회의원선거에서 SBS 방송국이 사용하였고, '96년 미국 대통령선거에서도 ABC 및 CBS 방송국 등이 사용하는 등 각종 선거방송에서 자주 사용되고 있다. 일본의 후지 TV의 메인 뉴스 '슈퍼타임'에서는 '95년 12월부터 일기 예보 방송에 가상 스튜디오 시스템을 도입하여 월요일부터 금요일까지 주 5일간 생방송을 하고 있어, 위성 TV 등을 통하여 수시로 접할 수 있다. 현재 고급 시장은 RT-SET, Accom 사 등에서, 중급 시장은 Digimedia 사에서 보급을 많이 하였으며 세계적으로 약 100여 시스템 내외가 보급되었으며 중급 시장에 주력하는 Digimedai 사에서는 30 시스템 이상을 전세계에 공급하였다. 향후 기술의 발전과 시장이 확대되면 현재의 저급 시스템 가격으로 고급 시스템 이상의 기능을 제공하게 될 것이다. 국내 시장의 규모는 약 100 시스템 정도로 추정된다. 또한 아직 관련 소프트웨어에 비하여 하드웨어의 가격이 비싸며 소프트웨어의 성능도 벤더에 따라 주력하는 방향이 다르므로, 방송사에서는 여러 개의 소프트웨어를 복합적으로 사용하는 추세이다. 일 예로 아시아 TV는 '96 국회의원 선거 방송에서 이스라엘

<표 2> 상용 시스템

	제 작 사	제 품	특 징
1.	RT-SET (이스라엘)	Laurus & Otus	• 그림자, Prompt, Preview 등 각종 기능을 충실하게 사용
2.	Digimedia/Brainstorm (스페인)	Multimedia ESTudio	• 최초로 Onyx 2 사용 • 위치이동 보조장치 등 사용
3.	For-A Orad	Virtual Set Cyber Set & Digiwarp	• 패턴 인식 방법 • 가상 인물 기능 구현
4.	Accom (미국)	Elset	• 저가격의 NT 계열 시스템 발표
5.	Discreet Logic (캐나다)	Vapour & Glass	• 3차원 모델링 toolset 제공
6.	Evans+Sutherland (미국)	MindSet	• 본격적인 NT 계열 시스템 개발

의 Orad 및 스페인의 Brainstorm을 사용하였다. 국내에서는 KBS 기술연구소에서 자체 개발한 'DreamSet' 및 MBC에서 한국과학기술연구원과 공동 개발한 '매직 스튜디오 III' 시스템 등을 보유하고 있으며, 선거 및 각종 방송물의 제작에 수시로 사용하고 있다. 현재 한정된 분야에서 사용되고 있는 가상 스튜디오 기술은 향후 각종 연예, 오락, 및 드라마 등의 방송 프로그램에 보편적으로 사용될 기술의 하나가 되고, 가상 스튜디오야말로 TV 방송국에서 가장 많이 보유하게 될 스튜디오 시설 중의 하나가 될 것이다. 활발한 사용을 위하여 하드웨어 및 소프트웨어 시스템의 구비뿐만 아니라, 프로그램 제작을 위한 전용 스튜디오 등의 설비, 입체감, 조형미 및 심도 있는 배경 제작을 위한 전문 디자이너, 각종 기능의 이해 및 활용이 가능한 PD 및 가상 스튜디오 제작 기술을 가진 엔지니어 등이 양성되어야 한다. KBS 기술연구소는 이러한 각종 연구를 추진하며 관련자에게 해당 기술력의 보급을 통하여 가상 스튜디오 시스템 확산에 노력하고 있다.

3. DreamSet 시스템

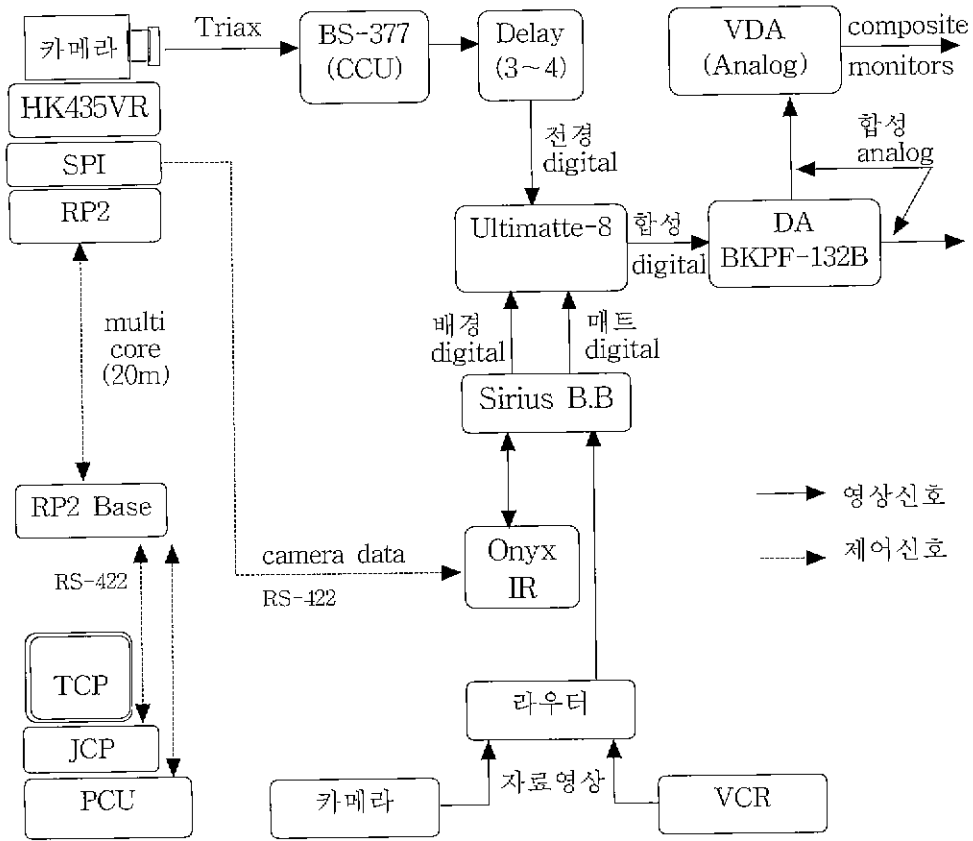
KBS 기술연구소에서는 '96년에 가상 스튜디오 연구를 시작하였다. 현재는 기본적인 기술 개발을 완료하였으며, 15대 대통령 선거 방송 등에 사용하였다. 그림 1은 가상 스튜디오 시스템의 기본 구성도이다. 시스템 1세트가 추가로 설치하여, 메인과 백업 개념으로서의 완벽한 시스템을 갖추었으며, 향후 2대의 카메라를 사용하여 cut과 fade 조작을 할 수 있는 시스템 개발의 기반을 마련하였다.

전체적인 시스템은 내부적으로는 CCIR-601의 디지털 신호를 처리할 수 있도록 구성하였으며, 최종단에 아날로그 시스템을 지원하기 위한 DA

컨버터를 장착하였다. 전경 영상 촬영을 위한 카메라는 Ikegami사의 HK377P와 HK388PW를 사용하였다. HK377P에는 base station에 디지털 비디오 옵션을 장착하여 CCIR-601의 Serial Component Digital 신호를 출력할 수 있도록 하였으며, 금년에 도입된 HK388PW 카메라의 경우 디지털 비디오 유닛이 기본으로 장착되어 있다. 이들 카메라는 카메라 데이터 추출장치가 포함된 Radamec사의 RP2 로보틱 페디스탈에 장착되어, 원격 또는 수동 제어가 이루어진다.

카메라로부터 출력된 비디오 신호는 디지털 프레임 딜레이 장치를 경유하여 크로마키어 장비에 입력된다. 딜레이를 주는 이유는 카메라 데이터가 전송되어 그래픽 워크스테이션에서 배경 영상을 생성하는데 소요되는 3-4 프레임 정도의 지연을 보상하기 위한 것으로 Sony사의 디지털 비디오 I/F 유닛을 사용하였다. 딜레이를 사용하지 않을 경우 합성된 화면에서, 배경이 전경 영상보다 늦게 움직이게 되어 부자연스러운 화면이 출력된다. Sony사의 BKPF 시리즈의 디지털 비디오 I/F 유닛은 모듈별 구성이 가능하도록 설계되어 있다. 본 시스템에서는 8개의 딜레이 유닛을 장착하여, 하나의 시스템당 4 frame의 딜레이를 주었으며, 별도로 DA 보드와 디지털 비디오 분배 보드를 장착하여 전체 시스템을 구성하였다.

배경 영상을 생성하는 그래픽 워크스테이션으로는 Silicon Graphics(SGI)사의 Onyx IR을 사용하였다. Onyx IR은 Radamec 장비로부터 RS-422 신호로 카메라 움직임 데이터를 입력받아 초당 60 프레임으로 배경 영상을 발생시킨다. 60 필드가 아닌 60 프레임을 사용하는 이유는 Onyx IR이 필드 단위로 그래픽을 생성하는 기능이 없기 때문이다. 60 프레임 rate으로 생성된 그래픽은 Onyx IR의 그래픽 출력 장비인 시리우스 보드를 통하여, 60 필드의 NTSC 규격에 맞게 출력된다. Onyx IR을

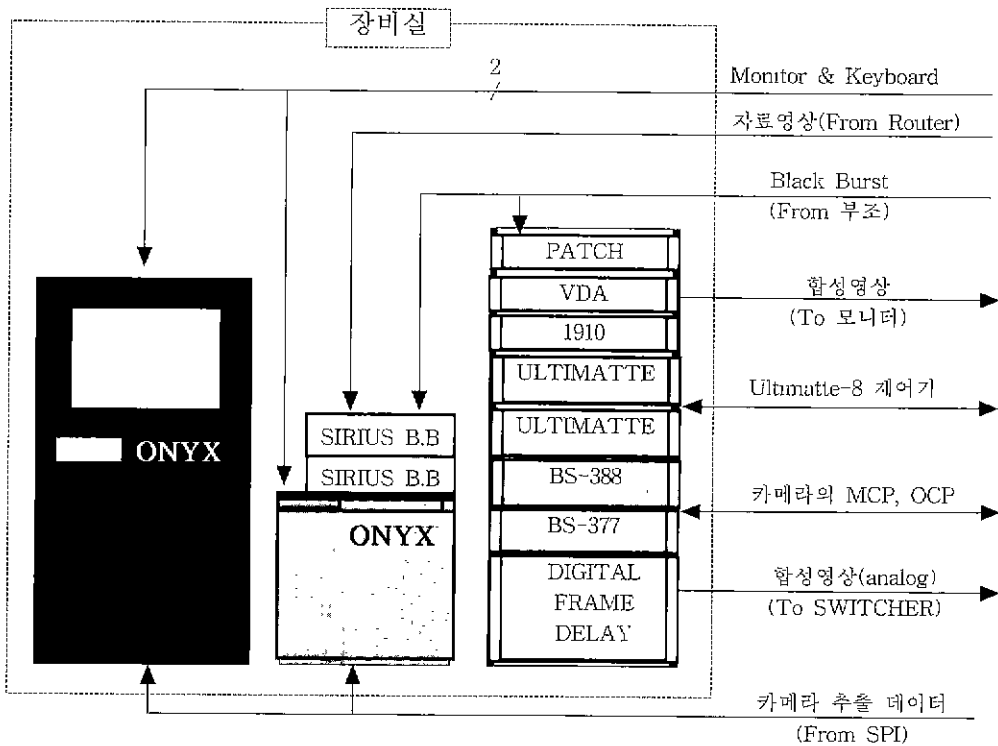


(그림 1) 가상 스튜디오 시스템 구성도

30 프레임 rate으로 설정할 경우, 카메라 움직임에 따른 배경 영상의 움직임이 자연스럽지 못하다.

크로마키어 장비는 카메라로부터 입력된 전경 영상과 그래픽 워크스테이션으로부터 발생된 배경 영상을 합성하는 장비로서 Ultimatte사의 Ultimatte-8을 사용하였다. Ultimatte-8은 디지털 비디오를 근간으로 구성되어 아날로그 비디오를 처리하는 기능이 없다. 본 시스템에서는 대부분의 장비가 디지털 비디오를 지원하므로 문제가 발생하지 않았으며, 다만, 기존의 스튜디오가 대부분 아날로그 장비로 되어 있으므로, 이를 지원하기 위해 위

에서 언급한 바와 같이 Sony사의 DA 컨버터 유닛을 사용하였다. Ultimatte-8은 본체와 제어기로 분리되어 구성되는데, 본체와 제어기는 RS-422 라인을 통하여 연결된다. 그런데, 현재 Ultimatte-8의 결함으로 인하여, Power-On시 본체와 제어기간의 통신이 이루어지지 않는 경우가 가끔 발생하는데, 이를 해결하기 위해서는 Ultimatte-8 본체의 디지털 Reference 입력을 항상 연결하여야 한다. 본 시스템에서는 카메라의 디지털 출력을 연결하여 이 문제를 해결하였다.



(그림 2) 가상 스튜디오 장비실 구성도

그림 1에서와 같이 가상 스튜디오 시스템은 합성된 최종 출력영상을 얻기 위해 카메라와 그래픽 워크스테이션으로부터는 영상신호를 입력받고 카메라 데이터 추출장치로부터는 위치 데이터와 같은 제어신호를 입력받는다.

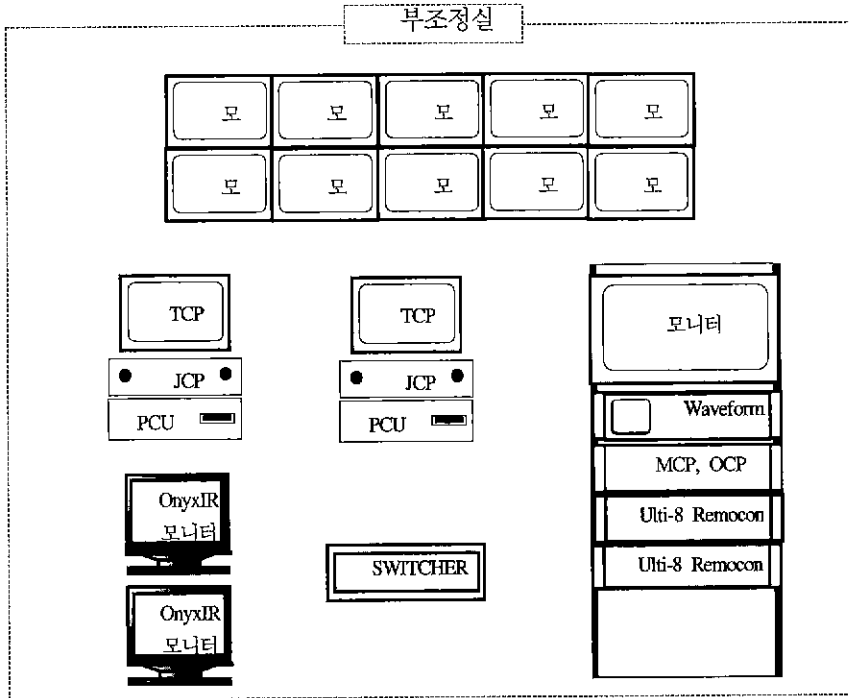
일반적인 TV 스튜디오는 부조정실과 실제 스튜디오의 두 장소로 나누어 진다. 가상 스튜디오도 마찬가지로 부조정실과 스튜디오가 필요하며, 별도로 그래픽 워크스테이션을 설치하기 위한 장비실이 필요하다. 그래픽 워크스테이션은 15~35°C의 온도를 유지시켜 주어야 하며, 자체 팬에 의한 소음이 문제가 되므로, 방음과 환온, 항습 기능이 구비된 장소에 설치하여야 한다. 장비실에는 그래픽 워크스테이션과 Ultimatte-8 본체, 카메라 CCU, 프레임 딜레이 등을 설치한다. 그림 2는 장비실의

구성도를 나타낸다.

부조정실에는 각종 장비를 제어하고 모니터링하기 위한 장비를 설치한다. 즉, 그래픽 워크스테이션의 모니터와 키보드, Ultimatte-8의 원격 제어 장치, CCU 제어를 위한 MCP, OCP 등을 부조정실에 설치한다. 로보틱 페디스탈을 조정하기 위한 PCU, JCP, TCP 등은 스튜디오 내에 설치하는 것도 가능하지만, 부조정실에 설치하는 것이 타당하다. 카메라맨은 부조정실 상단에 설치된 모니터를 통하여 합성된 영상을 볼 수 있으며, TCP를 이용하여 RP2를 제어한다. 이때, RP2의 위치 및 각도 조정에서 약간의 오차가 발생하므로, JCP의 joystick을 이용하여, 수동으로 이를 보정해주는 것이 필요하다. PCU, JCP는 스튜디오 내에 있는 RP2 base station과 RS-422로 연결되며, RP2-base

station은 multicore 케이블을 통하여 RP2와 연결된다. Multicore 케이블은 10m 길이 2개를 직렬로 연결 가능하므로 20m 범위로 RP2를 이동시킬 수 있다. 그림 3은 부조정실의 구성도이다.

즈의 초점거리도 스튜디오의 크기를 결정하는 중요한 요소가 된다. 큰 가상 세트를 연기와 합성하기를 원할 경우, 상대적으로 연기는 작은 크기로 촬영되어야 한다. 즉, 그만큼 카메라를

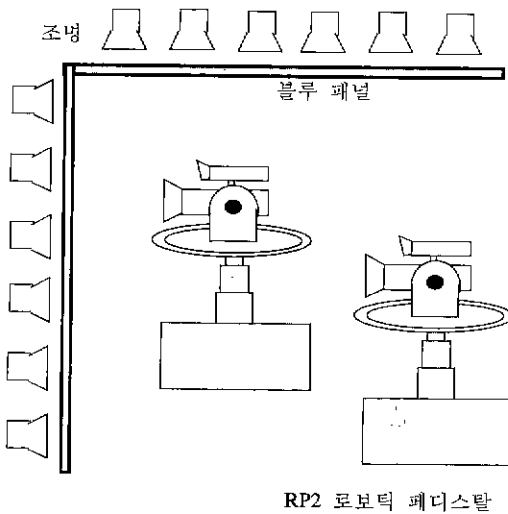


(그림 3) 가상 스튜디오 부조정실 구성도

스튜디오의 크기는 카메라가 촬영하는 전경 영상의 크기 및 이동 반경에 의해 결정된다. 가상 스튜디오의 주요 목적중 하나가 실제 세트의 제작 비용을 줄이는 것이므로, 스튜디오의 크기를 결정짓는 것은 주로 연기자의 이동 범위이다. 즉, 보도 프로그램과 같이 연기자가 고정된 장소에 위치하는 경우는 작은 스튜디오만으로 충분하며, 반면 연기자가 가상 세트 내에서 이동하는 효과를 주기 위해서는 연기자의 이동 범위에 해당하는 스튜디오 크기를 확보해야 한다. 카메라 렌

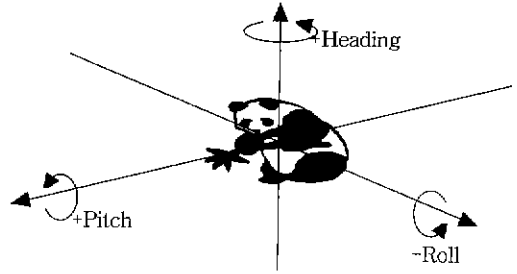
뒤에 위치시키는 것이 필요하며, 이것은 좀 더 큰 스튜디오를 요구한다. 이를 해소하기 위해서는 광각 카메라 렌즈를 사용하는 것이 필요하다. 현재 본 연구소에서는 Fuzinon사의 8-120mm 렌즈를 사용하는 데, 작은 스튜디오에 적합하도록 하려면, 5mm 이하를 지원하는 광각 렌즈를 채용하는 것이 요구된다. 스튜디오의 크기가 결정되면, 그에 해당하는 블루 패널을 설치한다. 블루 패널의 경우 Ultimatte사에서 권고하는 정확한 색상을 사용하는 것이 합성 화질의 질을 높일 수 있다. 또

한, 블루 패널에 균일한 조명을 비추는 것이 필요하며, 이를 위하여, 블루 패널의 각 모서리를 꼭면으로 처리하는 것이 권장된다. RP2를 가상 스튜디오의 카메라 추출 장치로 사용할 경우 RP2가 이동하는 스튜디오 바닥을 편평하게 제작하여야 한다. 그렇지 않을 경우, 카메라의 흔들림이 카메라가 촬영한 전경에 영향을 주게 되어, 합성 화면이 흔들리게 된다. 그림 4는 스튜디오의 구성도이다.

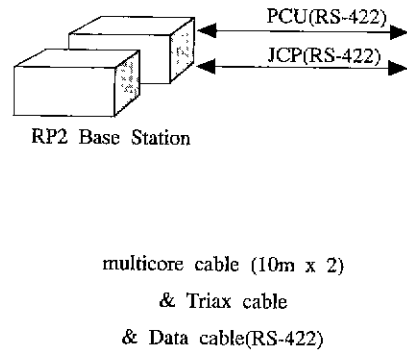


(그림 4) 가상 스튜디오의 스튜디오실 구성도

패러미터와 시점에 의하여 보는 방향이 결정되며, 그림 6에서 도시한 FOV에 의해 시야각이 결정된다.



(그림 5) 회전에 의한 가상 카메라의 설정 요소

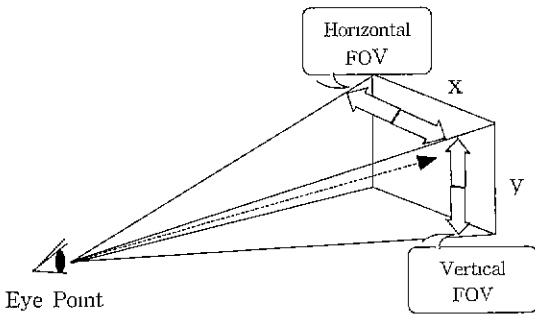


4. 카메라 캘리브레이션

그래픽 워스테이션에서 생성하는 배경영상은 카메라에서 촬영한 전경 영상에 동기되어야 한다. 즉, 카메라의 pan, tilt, zoom, focus와 x, y, z 움직임에 따라 생성되는 배경이 함께 연동되어 움직여야 한다. 이를 위하여, 전경을 촬영하는 실제 카메라와 배경을 촬영하는 가상 카메라를 일치시키는 작업이 필요하다. 이러한 작업은 가상 스튜디오에서 제일 먼저 해야하는 중요한 연구이다. 가상 카메라는 그림에서 도시한 heading, pitch, roll

Heading은 카메라의 pan에 해당하며, pitch는 tilt에 해당한다. Roll은 카메라가 옆으로 회전하는 것을 의미하는 데, 현재 본 연구소에서 사용하고 있는 RP2 로봇틱 페디스탈에는 roll 기능이 없으므로 사용하지 않는다. 다만, 실제 카메라와 연동하지 않고, 가상 카메라 자체만 rolling하는 효과를 주기 위해서는 사용할 수 있다. FOV(Field Of View)는 시야각을 나타낸다. 시야각은 카메라의 zoom 과 focus에 의해 결정된다. 즉, 카메

라를 zoom in하면 FOV는 작아지고 zoom out하면 커지게 되며, Focus 조정시에도 FOV가 변화한다. 사용하는 카메라의 aspect ratio를 알면, 수직, 수평 FOV 중 하나만 설정하여 사용하는 것이 가능하다.



(그림 6) 가상 카메라의 FOV

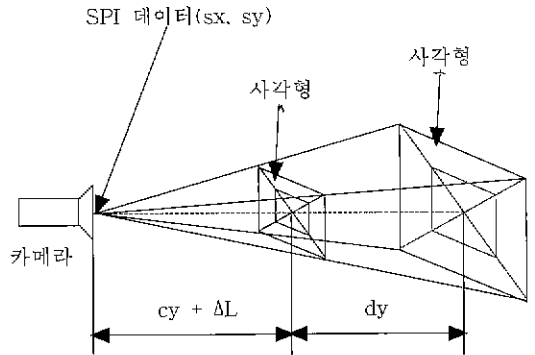
수식에서의 같이 Aspect Ratio(화면비)는 수평화소와 수직화소의 화면비율을 나타낸다.

$$Aspect\ Ratio = \frac{y}{x} = \frac{\tan(vertical\ FOV/2)}{\tan(horizontal\ FOV/2)}$$

카메라 캘리브레이션은 위와 같이 가상 카메라를 결정짓는 파라미터를 추출하는 과정이다. 카메라 캘리브레이션에서 가장 어려운 작업은 렌즈 캘리브레이션이다. 카메라 렌즈는 광학적으로 비선형 특성을 나타내므로 이를 보정하기 위해서는 정밀한 측정이 필요하다. 여기서는 RP2 로보틱 페디스탈을 기준으로 카메라 캘리브레이션을 설명한다. RP2에 부착된 SPI(Serial Position Interface)는 zoom position 데이터를 3 byte로 전송한다. 이 데이터는 Power-On이나 SPI reset시 변화하기 때문에, 처음 기동시에 최대 zoom out을 행하여, 이 값을 '0'로 지정하였다. 그 다음에 렌즈의 비선형 특성에 의한 카메라 시점의 변화를 보정하

기 위하여 다음과 같은 실험을 행하였다.

먼저 크기와 색상이 다르고 얇은꼴인 사각형 2개를 투명한 용지에 그렸는데, 사각형 A와 사각형 B의 비율은 1:2가 되도록 하였다. 이 사각형을 그림 7과 같이, 두 사각형간의 거리(dy)와 사각형 A와 임의로 정한 카메라 렌즈의 기준 위치와의 거리(cy)가 똑같이 되도록 배열한 후, SPI 데이터(six, siy)를 기록하였다. 위의 준비 작업을 거친 후, zoom in에서 zoom out까지를 13 단계로 나누어, 각각의 zoom 값에 대하여, 사각형은 고정하고 카메라를 앞뒤로 이동시켜 사각형 A와 사각형 B가 정확히 일치하는 점에 멈춘 후, SPI 데이터(sx, sy)를 측정하였다.



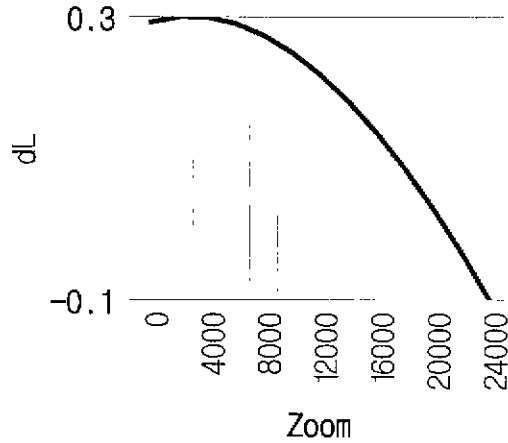
(그림 7) 렌즈 특성에 의한 시점의 이동 측정

측정한 데이터를 이용하여, 각 zoom 값에 대한 시점의 이동 정도(Δy)를 구하였다. 즉,

$$\Delta L = sy - siy \text{ 이며, 그림 8은 측정 결과를 나타낸다.}$$

위의 측정 데이터를 프로그램에 반영한 후, 렌즈 특성에 따른 FOV의 변화를 측정하기 위한 실험을 행하였다. 임의의 크기의 직사각형 종이를 제작하여, 카메라 앞에 위치시킨 후, 카메라 렌즈의 기준 위치와 종이와의 거리를 측정하였다. 그 다음 프로그램에서 실제 종이 크기와 같은 직사각형의 그래픽을 카메라 시점을 기준으로 실제 종이 위치에 발생시킨 후 이를 크로마키어로 합성하여, 크기 비교를 행하였다. 그림 9는 FOV의 측정과정을 나타낸다.

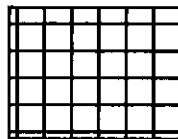
Zoom	ΔL
0	0.3138
2,000	0.3229
4,000	0.3218
6,000	0.3111
8,000	0.2914
10,000	0.2632
12,000	0.2268
14,000	0.1828
16,000	0.1312
18,000	0.0723
20,000	0.0063
22,000	-0.0667
24,000	-0.1467



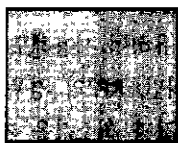
(그림 8) 렌즈 특성에 따른 카메라 시점의 이동 측정 데이터그림

그림 9에서 가상 물체와 실제 물체는 위치와 크기가 같으므로 합성된 화면에서 일치해야 한다. 따라서, (a)와 같은 상황이 발생한 경우는 가상 카메라의 FOV를 작게 조정하고, (b)와 같은 상황에서는 가상 카메라의 FOV를 크게 하여, 두 물체를 일치시키도록 하여야 한다.

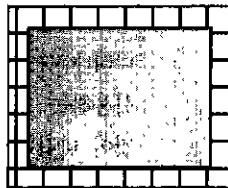
두 물체가 일치되면, 그 때의 SPI의 zoom, focus 데이터와 가상 카메라의 FOV 값을 기록한다. 이러한 과정을 13 단계의 zoom 값에 대하여 측정하였으며, 다시 각각의 zoom 값에 대하여, focus를 13단계로 나누어 FOV 데이터를 측정하였다. Zoom과 focus 데이터 측정시 focus에 따라 전경



실제 물체

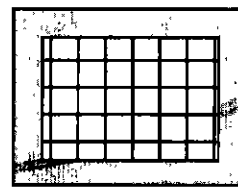


가상 물체



(a)

가상 카메라의 FOV가 실제보다 크게 설정된 경우



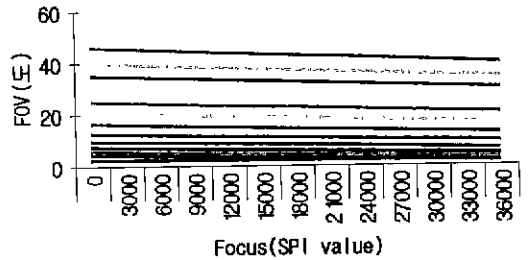
(b)

가상 카메라의 FOV가 실제보다 작게 설정된 경우

(그림 9) FOV의 측정과정

영상의 초점이 안 맞는 경우가 발생한다. 특히, zoom in 상태에서는 전경 영상의 경계를 뚜렷이 볼 수가 없어서 합성 영상에서 실제와 가상 물체를 일치시키는 작업을 하는 것이 어려웠다. 이러한 이유로, 측정된 데이터를 나중에 그래프로 그려서 비연속적인 점을 보정하여 데이터를 수정하였다.

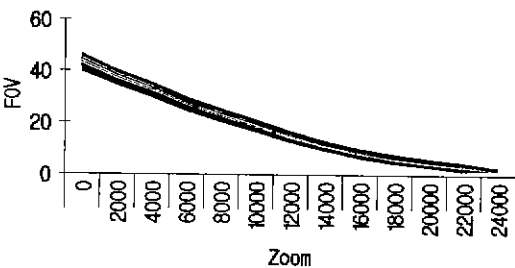
표 3은 SPI의 zoom, focus 데이터를 변화시켜가며 측정한 FOV 값을 나타내며, 그림 10은 이를 zoom 값의 변화를 기준으로 하여 그래프로 도시한 것이다.



(그림 11) SPI의 focus, zoom 데이터 변화에 따른 FOV 측정값

<표 3> DPI의 zoom, focus 데이터 변화에 따른 FOV 측정값

Focus Zoom	0	3,000	6,000	9,000	12,000	15,000	18,000	21,000	24,000	27,000	30,000	33,000	36,000
0	45.9070	45.4069	44.9068	44.4069	43.9070	43.4070	42.9070	42.5070	42.1067	41.6070	41.1070	40.6070	40.1070
2,000	39.7732	39.3415	38.9109	38.4807	38.0506	37.6040	37.1575	36.7340	36.3105	35.9327	35.5550	35.1773	34.7996
4,000	35.0019	34.5987	34.1956	33.7967	33.3977	33.0244	32.6511	32.2219	31.7927	31.3949	30.9971	30.5993	30.2014
6,000	29.3948	29.0177	28.6406	28.2706	27.9006	27.6002	27.2999	26.8620	26.4211	26.0620	25.6999	25.3378	24.9757
8,000	24.8699	24.5130	24.1561	23.7561	23.3561	23.0897	22.8232	22.4479	22.0726	21.7402	21.4078	21.0754	20.7430
10,000	20.7397	20.4245	20.1093	19.7101	19.3109	19.0088	18.7068	18.3578	18.0089	17.7599	17.5109	17.2619	17.0129
12,000	16.3775	16.0621	15.7468	15.4219	15.0970	14.8055	14.5140	14.2657	14.0174	13.7390	13.4605	13.1821	12.9036
14,000	12.5198	12.2967	12.0736	11.8097	11.5459	11.3114	11.0770	10.8010	10.5250	10.3439	10.1628	9.9816	9.8005
16,000	9.5943	9.3979	9.2016	8.9672	8.7328	8.5099	8.2870	8.0870	7.8870	7.6209	7.3549	7.0888	6.8227
18,000	7.3448	7.0994	6.8541	6.6162	6.3783	6.1783	5.9783	5.7783	5.5783	5.1783	4.9783	4.7783	
20,000	5.5599	5.4060	5.2521	5.0525	4.8530	4.6764	4.4999	4.3237	4.1596	3.9598	3.7599	3.5601	3.3602
22,000	4.2219	4.0590	3.8962	3.7195	3.5428	3.3721	3.2014	3.0014	2.8014	2.6514	2.5014	2.3514	2.1814
24,000	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749	2.2749



(그림 10) SPI의 zoom, focus 데이터 변화에 따른 FOV 측정값

이상과 같은 과정을 거쳐 카메라 캘리브레이션이 끝나게 된다. 근본적으로 동일한 모델의 렌즈라 해도 제작상의 오차에 의해 모든 렌즈가 같은 특성을 나타내지는 않는다. 따라서, 정확한 사용을 위해서는 사용하는 모든 렌즈에 대해 캘리브레이션을 행하는 것이 바람직하지만, 실험결과 동일한 모델의 경우 하나의 측정 데이터를 공유하여 사용해도 크게 무리가 없음이 확인되었다. 다만, RP2의 경우 zoom, focus, pan, tilt에 대하여 최대값과 최소값을 지정하는 옵션을 제공하는데, 이 설정이 서로 다를 경우, 문제가 발생할 소지가 있으므로 주의가 요망된다.

5. 소프트웨어 개발

최근에 개발된 가상 스튜디오 프로토타입은 Performer 라이브러리만 구성되었다. Performer 라이브러리로만 GUI 기능을 확대시키기 어렵기 때문에 Motif 라이브러리와 결합하여 개발한 후 보다 다양한 애니메이션의 필요성 때문에 ViewKit 라이브러리로 애니메이션 스크립터를 개발하여 클라이언트 서버 형식으로 개발되었다. 클라이언트 서버 형식의 장점은 두 대 이상의 카메라와 두 대의 ONYX 컴퓨터가 같이 사용될 때 서버 프로그램의 변경 없이 데이터를 처리하는 클라이언트 구조의 변경만으로 쉽게 연결 사용할 수 있을 뿐 아니라 프로그램 모듈화로 효과적인 개발과 사용환경을 제공할 수 있다는 것이다. 단점으로는 공유될 데이터의 통신 인터페이스 문제, CPU Locking 문제 등이 있다. DreamSet을 운영하기 위한 서버 프로그램과 애니메이션 스크립터 이외에 세트를 제작하기 위해서 Z-Y 변환기, SDL-LP 변환기, SDL-IV 변환기와 애니메이션 스크립터에 쉽게 모델링 소프트웨어에서 제작한 애니메이션을 추가하기 위한 애니메이션 변환기도 개발하였다.

서버 프로그램은 Performer 라이브러리를 사용하여 실시간 그래픽 처리를 한다. Motif 라이브러리는 당초 애니메이션을 비롯한 환경 설정 등 다양한 GUI를 제공하기 위해서 사용하였는데 ViewKit 라이브러리를 사용한 애니메이션 스크립터를 개발한 이후 애니메이션과 관련된 기능이 필요없게 되었다. 현재는 환경 설정에 필요한 카메라 설정, 블루 패널 설정, 스튜디오 위치 설정 등과 같은 GUI를 제공하기 위해 Motif 라이브러리를 사용하고 있다. 추후 환경 설정 프로그램이 완성된 후 Motif 라이브러리를 사용하지 않고 Performer 라이브러리만 사용하여 서버 프로그램의 안정도와 크기를 줄일 예정이다.

본 프로그램에서는 다양한 애니메이션과 뛰어난 화질 구현에 초점을 두었다. 다양한 애니메이션을 구현하기 위해서는 프리미티브 기능의 설정이 중요하다. 구현된 프리미티브 기능으로는 이동, 회전, 크기 변경, Depth Key, Matte 기능, Fade, 텍스처 변환, 숫자 표시 등이 있다. 애니메이션을 구현할 때 중요한 것중에 하나는 바로 속도감이다. 특히 애니메이션 시작때와 끝날때의 가속과 감속은 애니메이션을 부드럽게 하여 자연스럽게 한다. 이러한 기능을 위해서 가속구간, 감속구간 설정, 반복 횟수 기능 등을 구현하였다. 또한 보다 다양한 애니메이션 구현과 모델링 소프트웨어로부터의 애니메이션을 도입하기 위하여 키프레임마다의 접선벡터 설정 공간을 설정하였다. 각 애니메이션은 키프레임과 접선벡터 설정에 의한 SPLINE 방식에 의해 계산되어 실행하도록 했다.

특히 애니메이션중 방송에 꼭 필요한 애니메이션이 바로 텍스처 애니메이션과 숫자 표시이다. 텍스처 애니메이션은 텍스처 애니메이션 스크립트 파일을 제작하여 여러 가지 텍스처를 모델에 다양하게 입힐 수 있다. 숫자 표시의 경우에는 숫자 디지털에 맞게 모델을 제작하면 자연수나 소수에 상관없이 표현이 가능하다. 특히 자연수의 경우 세자리마다 자동으로 콤마(,)를 삽입할 수 있어서 선거방송과 같이 숫자 데이터가 특히 중요한 경우에 꼭 필요한 기능이다.

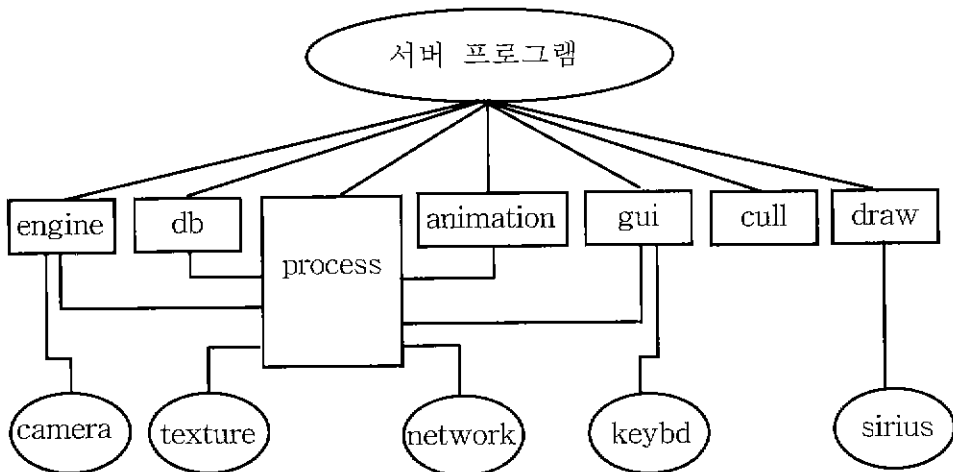
뛰어난 화질을 구현하기 위해서는 보다 많은 시간과 메모리량을 필요로 한다. 보다 자연스러운 그래픽 품질을 내기 위해서 3차원 데이터 로더를 개선하였다. 특히 텍스처가 색분리되는 것을 막기위해서 16비트 텍스처에서 32비트 텍스처로 모두 바꿈으로써 사람의 눈으로는 인접색의 차이를 느끼지 않고 자연스럽게 표현되게 하였다. 그러나 이 작업은 2배의 텍스처 메모리와 그리

는 시간을 더 요구한다. 이를 위해 복잡한 세트의 경우 최적화 작업을 하였다. 특히 같은 재질의 경우 한 그룹으로 처리하여 재질 셋팅에 따른 속도 저하를 방지하였고, 그 외에 필요 없는 트랜스폼 노드나 기타 여러 가지 노드들을 삭제함으로써 속도 향상과 메모리 절약을 하였다. 애니메이션되는 오브젝트의 경우 필요한 경우에만 pivot을 설정하게 함으로써 pivot 설정에 따른 불필요한 SCS 노드를 없앴다. 멀티 프로세싱에 따른 메모리 공유와 불필요한 메모리 설정을 최대한 없애기 위하여 모듈별로 Initialize 함수와 End 함수를 설정하여 멀티 프로세스 설정이 끝난 후 메모리 설정이 되도록 구현하였다.

본 프로그램은 크게 engine 모듈, db 모듈, process 모듈, animation 모듈, gui 모듈, cull 모듈, draw 모듈로 나누어진다. 이중 앞에 5개의 모듈은 어플리케이션 프로세스에서, cull 모듈은 쉘 프로세스에서 draw 모듈은 드로우 프로세스에서 실행된다. engine 모듈은 Performer 라이브러리를 초기화하고 카메라 입력을 받아서 그래픽을 생성한다.

db 모듈은 그래픽 데이터를 생성한다. process 모듈은 모든 명령어를 받아서 animation을 설정하거나 환경을 설정한다. animation 모듈은 앞에서 설명된 여러 가지 애니메이션을 처리한다. gui 모듈은 Motif 라이브러리로 작성되었고 버튼 입력을 받아 명령어 형태로 process 모듈로 전달한다. cull 모듈은 cull 기능에 관련된 callback을 실행한다. draw 모듈은 시리우스 입출력, 환경설정, 확장 블루 패널 기능 등과 draw에 관련된 callback을 실행한다. 각각의 모듈에는 서브 모듈이 존재할 수 있으며 이를 그래프로 도식하면 그림 12와 같다.

animation 모듈은 모델에 관한 애니메이션을 처리한다. 크게 두가지 함수들로 나뉘어져 있는데 애니메이션 설정 부분과 애니메이션 처리 부분이다. process 모듈과 같이 animation 모듈도 애니메이션 스크립터 개발 전 만들어진 옛날 함수들과 새로운 함수들이 공존한다. 옛날 함수들은 시작점과 끝점의 애니메이션 값과 시작 시간, 끝 시간의 시간 값, 반복 횟수, 애니메이션 가속/감속 구간, 애니메이션 cull 설정 등을 할 수 있다. 새로운 함수는 시간 값과 애니메이션 값을 하나의 벡터로



(그림 12) 서버 프로그램 모듈 관계도

처리하여 멀티 벡터 애니메이션이 가능하다. 설정할 값에는 반복횟수와 애니메이션 벡터만 설정하면 되어 옛날에 비해 설정, 기능, 성능면에서 우수하다.

옛날 함수에서 애니메이션 설정 후 n 번째 프레임의 애니메이션 값은 식과 같다. 이 식은 가속/감속 구간을 선형으로 계산하였다. 시작 프레임 n_0 로 끝 프레임을 n_1 , 가속/감속 구간을 a , 시작점을 v_0 , 끝점을 v_1 로 했을 때 n 번째 프레임의 애니메이션 값 $f(n)$ 은 다음과 같다.

$$f(n) = v_0 + \frac{(n - n_0)^2}{a} \times (v_1 - v_0)$$

$$(n_0 \leq n < n_0 + \frac{a}{2})$$

$$f(n) = v_0 + (n - n_0 - \frac{a}{4}) \times (v_1 - v_0)$$

$$(n_0 + \frac{a}{2} \leq n < n_1 - \frac{a}{2})$$

$$f(n) = v_0 + (n_1 - n_0 - \frac{a}{2} - \frac{(n_1 - n)^2}{a}) \times (v_1 - v_0)$$

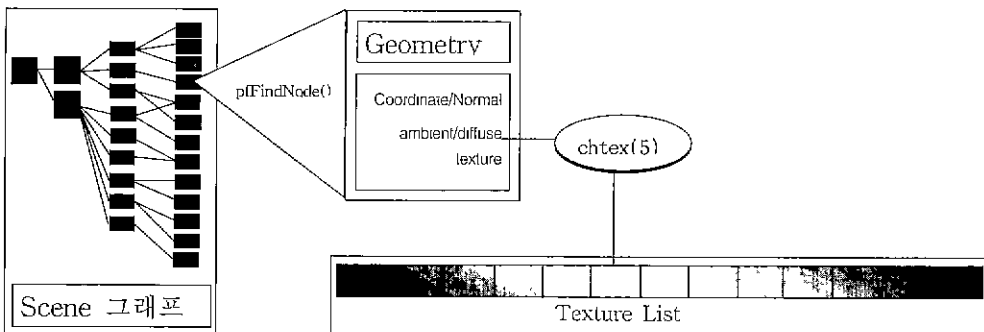
$$(n_1 - \frac{a}{2} \leq n \leq n_1)$$

새로운 함수에서는 하나의 벡터에 6개의 값을 가지게 된다. 대부분의 애니메이션의 경우 3개의 애니메이션 값을 필요로 한다. 처음 3개의 값은 애니메이션 값이고, 4번째 값은 시간, 5번째와 6번째의 값은 탄젠트 값으로 설정해놓았다.

탄젠트 값은 해당 프레임에서 들어오는 기울기와 나가는 기울기를 설정할 수 있다. 3개의 애니메

이션 값에 필요한 탄젠트값은 사실상 6개가 필요하다. 또한 3개의 값이 필요없는 애니메이션인 텍스처 애니메이션이나 숫자 애니메이션의 경우 2개의 값은 필요 없게 된다. 이러한 문제 때문에 애니메이션 설정 프로그램도 다분히 비효율적으로 작성되었다.

텍스처 애니메이션은 모델의 표면에 원하는 그림을 연속적으로 표시하는 기능이다. 3차원 애니메이션을 위해서는 미리 애니메이션 될 텍스처들을 지정하여야 한다. 이러한 설정 파일에 필요한 요소로는 애니메이션이 입혀질 표면의 geometry 이름, 입혀질 텍스처 숫자, 텍스처 이름이다. geometry는 세트 준비 과정에서 미리 이름을 정해 놓아야한다. 또한 다른 모델과 결합하여 사용하는 것을 피하도록 디자인한다. 애니메이션이 될 텍스처들은 <base 이름>+<일련번호>.rgba 식으로 저장되어야 한다. 이러한 과정을 거쳐 세트와 설정 파일이 제작되면 프로그램에서 세트에서 해당하는 geometry 이름을 찾아 자체 메모리에 텍스처 링크를 보관한다. 다음으로는 애니메이션될 텍스처 파일들을 읽어 텍스처 리스트를 제작한다. 애니메이션 명령어가 내려지면 텍스처 리스트로부터 해당 텍스처를 geometry 텍스처 링크에 삽입한다. 그림 13은 텍스처 애니메이션의 구조를 도식화한 것이다.



(그림 13) 텍스처 애니메이션 구성도

텍스처 애니메이션을 할 때 geometry 설정을 바꾸어 주어야 할 것들이 있다. 텍스처는 입혀지는 방식에 따라 DECAL, MODULATE, BLEND가 있는데, 순수 텍스처로 최종 출력을 원한다면 DECAL 방식이 좋다. 하지만 DECAL 방식은 투명도가 있는 텍스처의 경우 모델의 기본질감을 보여주기 때문에 특수한 경우에는 MODULATE 방식을 사용한다. MODULATE 방식의 경우에는 모델의 기본색을 하얀색으로 설정하여야 하고 투명도값을 0으로 설정하면 안되고 적은값이라도 설정되어야 한다. 이 문제는 Performer 라이브러리에서 투명도값이 설정되지 않았을 경우 속도향상을 위해서 새로 투명도값을 설정하지 않기 때문이다.

숫자 애니메이션의 원리는 텍스처 애니메이션과 동일하다. 단지 숫자 애니메이션의 경우에는 여러 자리수를 표시해야 하기 때문에 텍스처 애니메이션에서 사용하던 geometry 이름 대신에 여러 개의 geometry를 대표할 수 있는 베이스 이름을 사용하여야 한다. n개의 디지털을 사용하고 싶다면 베이스 이름에 0부터 n-1까지 숫자를 붙인 이름에 해당하는 geometry가 존재해야 한다. 또한 자리수를 표시하기 위해서 자동으로 3자리마다 콤마를 생성할 수 있다. 이 경우 콤마가 들어갈 geometry에는 베이스 이름에 com이라는 이름이 덧붙여지며, 콤마가 여러개가 들어갈 수 있기 때문에 0부터 n/3-1까지의 숫자가 붙게 된다. 소수점을 표시하기 위해서는 항상 소수점이 변하지 않는 부동소수점 방식을 사용한다. 텍스처 애니메이션과 숫자 애니메이션은 아직 미완성상태로 현재 개발되어 있는 것은 소수점 이하 한자리만 표시할 수 있도록 설계되어 있다. 소수점에 해당하는 geometry에는 베이스 이름에 dot이라는 이름이 덧붙여진다.

animation 모듈은 가상 스튜디오의 기능에서 핵

심적인 부분이다. 앞으로 개발될 대부분의 기능은 animation 모듈에서 이루어지게 된다. 현재로서는 초보적인 애니메이션만 가능하다. 앞으로 개발할 부분과 개선될 부분이 많다. 오브젝트 몰핑(Morphing) 기능은 그 활용할 수 있는 곳이 많다. 몰핑이란 것은 오브젝트의 위치나 크기가 변하는 것이 아니라 형태가 변화하는 것을 말한다. 현재 초보적인 몰핑 기능을 개발하였는데 Performer에 형태가 변함에 따른 cull 영역 설정이 되어있지 않아, 사용상 제약이 많다. 계층구조 애니메이션 기능은 로봇 팔과 같이 다관절로 이어진 시스템에서 상대적으로 애니메이션할 수 있는 것이다. 앞으로 가상 인물과 같은 애니메이션 팩터를 받아들이기 위해서는 두가지 방법이 있다. 계층구조 애니메이션을 가상 인물 모듈에서 계산하여 절대적 애니메이션을 보내주는 방법과 animation 모듈에서 자체 계산하는 방법이 있다. 서버 프로그램의 효율성과 작지만 강력한 기능을 위해서는 후자의 방법이 유리하다. 벡터 애니메이션을 개선하여야 한다. SPLINE 방식에 의한 계산을 위해서 벡터 컴포넌트의 구조를 애니메이션 값, 프레임, 입력 탄젠트, 출력 탄젠트의 형식으로 바뀌어야 한다. 이렇게 나누어졌을때의 문제점은 현재의 수치에 의한 애니메이션 스크립트 기록 방식으로는 애니메이션 구현이 어렵다는 것이다. 이 방법은 인터랙티브 애니메이션 에디터의 개발이 필수적이다. 텍스처 애니메이션은 현재 잘 구성이 되어 있다. 앞으로 자연스러운 텍스처 변환과 텍스처에 의한 포커스 IN/OUT 기능을 구현할 예정이다. 숫자 애니메이션은 현재 여러개의 geometry를 디자인 단계에서 구현해야 하기 때문에 많은 문제점을 안고 있다. 앞으로 하나의 geometry에 입힐 숫자 텍스처를 실시간으로 생성하는 방법으로 구현할 예정이다.

애니메이션 스크립터는 ViewKit 라이브러리와

C++로 제작되었다. C++은 구조체 프로그래밍과 최적화에 적합한 프로그래밍 언어로서 앞으로 모든 개발이 C++로 이루어질 예정이다. 애니메이션 스크립터는 스크립트 파일 해석 모듈, 사용자 인터페이스 모듈, 스크립트 실행 모듈, 외부 데이터 인터페이스 모듈로 나누어져 있다. 스크립트 파일 해석 모듈은 작성된 스크립트를 해석하여 내부 코드로 변환한다. 특히 수학 계산식을 사용할 수 있다는 것이 특징이다. 선거방송과 같이 수치 데이터를 그래프로 표시해야할 경우 수학 계산식은 반드시 들어가야만 한다. 수학 계산식은 후위연산자 방식으로 바뀌어진다. 후위연산자 방식은 연산 우선순위에 따라 연산자를 뒤에 배치하는 방식이다.

사용자 인터페이스 모듈은 ViewKit 라이브러리로 제작되었다. 페이지 개념으로 애니메이션을 설정할 수 있으며 각각의 페이지에 최대 24개의 버튼을 설정할 수 있다. 그리고 2대의 컴퓨터를 연결 사용하는 것을 고려하여 Local로 연결하는 것과 Remote Host로 연결하는 것에 대한 설정 박스를 만들었다.

스크립트 실행 모듈은 해석 모듈에서 만들어진 내부 코드 실행, 후위연산자 방식으로 만들어진 수학 계산식 계산, 외부 데이터 인터페이스에서 얻어진 데이터 처리, 내부 변수 사용, 애니메이션 스크립터 제어 등을 처리한다. 이러한 기능들은 선거방송에 꼭 필요한 기능으로 상용 가상 스튜디오 프로그램에서 쉽게 구현하기 어려운 부분이다. 내부 변수를 사용하였기 때문에 내부변수와 연동하는 애니메이션을 제작할 수 있다. 조건문장을 이용하여 경우에 따라서 애니메이션을 동작시킬 수가 있어, 스크립트만 잘 짜여지면 오퍼레이터가 여러개의 버튼을 누르지 않아도 되고 그만큼 오퍼레이터의 실수를 줄일 수 있다. 외부 데이터 인터페이스 모듈은 시리얼 포트를 통해

전해지는 선거 데이터를 받아 그 내용을 해석하고 저장한다. 이 모듈에서 얻어진 값은 문자열이나 실수 형태로 실행 모듈로 넘겨진다.

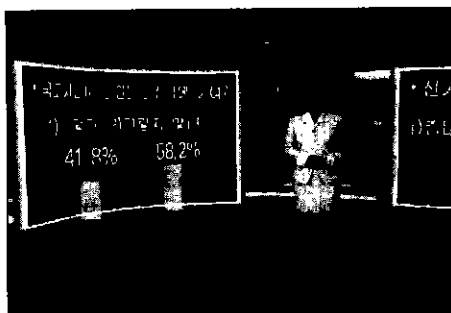
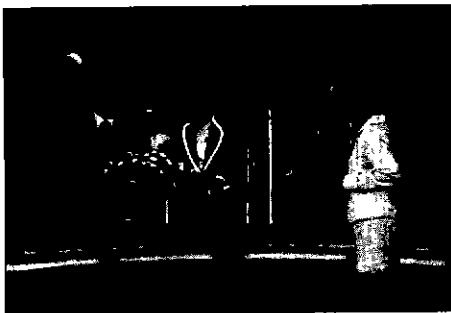
외부에서 3D 모델링한 파일을 DreamSet에서 사용하기 위하여는 여러 가지의 변환기가 필요하다. Alias사의 Power Animator에서 기본적으로 제공되는 Inventor 포맷은 Y-up이다. 이 때문에 Y-up을 Z-up으로 변환하는 과정이 필요한데, 서버 프로그램에서 3차원 모델을 읽고 X축으로 -90도 회전시켜도 되지만 이 경우 애니메이션이 복잡하게 된다. 그래서 제작된 프로그램이 Inventor 파일의 Y-up에서 Z-up으로 변환하는 Y-up Z-up 변환 프로그램이다. SDL IV 변환기는 SDL 파일을 Inventor 파일로 변환하는 프로그램이다. SDL 파일을 읽어서 새로이 그룹을 정하고 Indexed coordinate 시스템으로 적은 메모리를 요구하는 Inventor 파일로 저장한다. 애니메이션 변환기는 Power Animator에서 저장된 애니메이션 파일을 애니메이션 스크립터에서 읽을 수 있는 포맷으로 바꾸어준다. 이렇게 함으로서 디자이너가 애니메이션까지 모두 할 수 있다. 또한 물체가 떨어지는 시뮬레이션이나 인체가 움직이는 동작 등 복잡한 애니메이션까지 수용할 수 있기 때문에 훨씬 다양한 애니메이션을 보여줄 수 있다.

6. 결 론

제작비 절감 문제의 해결 혹은 실제 스튜디오에서 구현하기 어려운 배경 제작의 구현 등을 목표로 '94년부터 발표되기 시작한 가상 스튜디오는 '95년 초기 버전의 발표에서 '96년 새로운 기능의 추가 등을 통하여 현재 여러 방송국에 설치하여 프로그램 제작에 사용되고 있다.

KBS 기술연구소에서 '96년부터 시작된 가상 스튜디오 연구는 관련 기초 연구, 해외시장의 상

용제품 분석, 하드웨어 시스템의 구성 등 많은 성과가 있었다. '97년에는 한 발짝 나아가 이론적인 연구에서 실제 시스템 S/W의 프로그래밍이 진행되었다. 각종 시험 프로그램 및 대선방송 등의 제작 과정을 통해 가상 스튜디오에서 필요로 하는 요구사항을 파악하여 시스템 개발에 적용하거나 S/W에 존재하는 버그를 보완하였다. 금년 이후의 연구 추진 방향 결정에도 크게 도움이 될 것으로 생각한다. 그림 14은 시험 제작된 프로그램의 방송 화면을 보여 주고 있다.



(그림 14) 시험 제작된 프로그램의 방송 화면

앞으로 관련 분야의 발전 방향과 속도 등을 지속적으로 파악하며, 방송국 내의 도입, 설치 및 각종 선거에서의 활용 계획에 맞추어 가상스튜디오 운용 프로그램의 개발을 추진할 계획이다. 국내의 타방송사 및 학계에서 가상 스튜디오를 연구하는 곳이 있는 것으로 알고 있다. 서로간에

정보 및 연구 결과의 교환을 통하여 효율적으로 연구가 추진 되었으며 한다. 금번에 지면 관계상 카메라 캘리브레이션만 어느 정도 소개하였으며, 나머지는 개략적인 개변만 언급하였다. 차후 기회가 주어지면 자세하게 소개할 예정이다. 이러한 연구는 언젠가 가상 스튜디오의 목적인 경제적 프로그램의 제작과 환상적인 세트의 구성 등이 실현될 그날까지 계속 될 것이다.



최영진

1981년 한양대학교 전기과 (학사)
 1985년 한국과학기술원 전기전자 공학과 (석사)
 1998년-현재 KBS 기술연구소 차장



남승진

1989 연세대학교 전자공학과 (학사)
 1991 연세대학교 전자공학과 (석사)
 1998년-현재 KBS 기술연구소 연구원



최창락

1991년 한국과학기술원 전기 및 전자 공학과 (학사)
 1993년 한국과학기술원 전기 및 전자 공학과 (석사)
 1998년-현재 KBS 기술연구소 연구원