

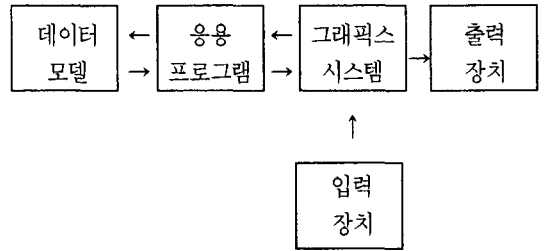
# 차세대 컴퓨터 그래픽스 기술의 전망

金昌憲

高麗大學校 컴퓨터學科

## I. 서론

컴퓨터 그래픽스는 여러 가지 물체와 경관, 또는 예술적인 영상을 컴퓨터를 이용하여 인공적으로 작성하는 기술이라 말할 수 있다. 컴퓨터 그래픽스는 그 이름에서 연상되는 것처럼 원래 컴퓨터의 도움을 받아 제도를 하는 CAD(Computer Aided Design)의 분야에서 발전하여 왔다. 그러나 컴퓨터의 성능 향상에 따라서 여러 종류 정보의 가시화나 예술 작품의 작성에도 컴퓨터 그래픽스 기술이 사용되어져 현재에는 컴퓨터 그래픽스라 함은 컴퓨터에 의한 영상 작성 전반적인 것을 의미하게 되었다.



〈그림 1〉 컴퓨터 그래픽스 시스템의 개념적 구조

프로그래머의 입장에서 본 컴퓨터 그래픽스 시스템은 그림 1과 같이 컴퓨터 그래픽스 장치, 컴퓨터 그래픽스 시스템, 응용 프로그램과 데이터 모델과 같은 네 가지 요소로 구성되어 있다<sup>[1]</sup>. 컴퓨터 그래픽스 장치는 컴퓨터 시스템과 함께 컴퓨터 그래픽스 하드웨어라고 볼 수 있고, 데이터 모델 및 응용 프로그램은 컴퓨터 그래픽스 소프트웨어이며, 컴퓨터 그래픽스 시스템은 컴퓨터 그래픽스 하드웨어와 소프트웨어를 연결하는 인터페이스 역할을 한다.

### 1. 형상 및 동작 모델링

데이터 모델과 응용프로그램은 컴퓨터 그래픽스 출력 이전의 문제로서, 컴퓨터 그래픽스에서 다룰 물체를 어떻게 모델링하여 컴퓨터 내에 표현할 것이며, 이것을 이용하여 어떤 일을 할 것인가가 그

이슈이다. 컴퓨터 그래픽스에서 다루는 물체는 다른 컴퓨터 응용 분야와는 달리 주로 물체의 기하학적 측면이 관심의 대상이다. 따라서 컴퓨터 그래픽스에서 데이터 모델이라 함은 물체의 형상을 다루는 기하학적 모델링과 그 표현 방법이라고 말할 수 있다. 컴퓨터를 이용한 자동 제도에서 시작한 CAD는 그 발전이 본격화됨에 따라, 보다 복잡한 물체를 효과적으로 모델링하기 위한 노력이 계속되어 왔으며, 1970년 말부터 상업용 솔리드 모델러(solid modeler)가 시장에 출현하기 시작했다<sup>[2,3,4,5]</sup>. 그리고, 컴퓨터 시스템의 눈부신 발달에 따라 보다 현실감 있는 물체의 모델링에 대한 인간의 도전은 계속되어 왔다. 즉, 다면체 모델링에서 곡면 모델링으로, 2차원에서 3차원으로, 정지 물체에서 연속적으로 움직이는 물체로 그 대상이 확대되어 왔다. 곡면의 정확한 표현은 그 자체로서 매우 흥미 있는 문제일 뿐만 아니라 많은 응용 분야에서 필요하다.

키-프레임(key frame)방식의 애니메이션은 물체의 움직임에 대한 역학적 요소를 무시하고 중요한 순간의 키-프레임, 즉 물체의 위치와 방향을 시간 순서로 나열하고 이것을 부드럽게 보간함으로써 물체가 자연스럽게 움직이는 것처럼 보이도록 하는 기법이다<sup>[6]</sup>. 물체의 초기 위치 및 방향과 속도를 주고 이후의 운동 궤적을 찾아 영상을 생성하는 운동학(kinematics)을 이용한 애니메이션은 키-프레임 방식보다 진보된 방법이다<sup>[7,8]</sup>. 물체의 초기 위치 및 방향과 속도만을 제어 변수로 하여 애니메이터가 원하는 운동 궤적을 구하기는 매우 힘들고, 어색한 움직임이 만들어 질 수 있다. 그래서 자연스러운 애니메이션을 위해서 역운동학(inverse kinematics) 및 동력학(dynamics)과 역동력학(inverse dynamics) 등을 이용한다<sup>[8,9]</sup>. 나무, 풀, 구름 및 산과 같은 자연물이나 구름, 불꽃, 안개와 같은 자연현상은 기존의 기하학적 모델링 방법으로는 거의 표현이 불가능하다. 프랙탈 기하학에 의하면 자연물이나 자연 현상은 매우 불규칙하게 보이지만, 자기 유사성이 있어서 많은 경우에 간단한 법칙으로 구성되는 동적 시스템(dynamical system)을 재귀적으로 적용하면 이들을 표현

할 수 있다. 1970년 말 Mendelbrot가 프랙탈 기하학을 제창한 이래, 기존의 기하학적 모델로서는 불가능했던 자연물이나 현상을 대상으로 한 환상적인 영상이 생성되어지고 있다<sup>[10,11,12]</sup>. 입자 모델링(particle modeling) 기법은 프랙탈 기법과는 달리 물체는 작은 입자들로 구성되며 입자 상호간의 작용에 의해 특정 순간의 물체의 모양이 결정된다는 매우 과학적인 이론에 근거하고 있으며, 입자와 상호 작용을 어떻게 정의하느냐에 따라 다양한 물체나 현상을 표현할 수 있는 기법이다<sup>[13,14,15]</sup>. 이 기법을 이용하여 폭포, 불꽃과 위성의 폭발 모습 등이 매우 현실감 있게 표현되어 그 유용성이 과시되었다. 그리고 주변 환경이 시간에 따라 변할 경우 형상도 따라서 변하는 물체를 표현하기 위하여 물체의 기본 형태를 작은 요소로 나누고, 이들 요소간에 상호 작용하는 힘과 외부에서 작용하는 힘이 균형을 이루는 방향으로 모습이 변해 가는 과정을 계산하여, 특정 순간의 물체를 결정한다. 이 방법은 기존의 형상 중심의 기하학적 모델과 물리적인 법칙을 결합하려고 하는 물리기반 모델링(physically-based modeling) 기법의 하나로 최근 많은 연구가 이루어지고 있다<sup>[8,16,17,18,19,20]</sup>. 이 밖에도 공상 과학 영화나 피기 영화에서 레이저 광선 총을 맞는 사람이 서서히 재로 변한다던가 사람이 괴물로 변하게 하는 메타머피시스(metamorphosis)와 어떤 모습에서 완전히 다른 모습으로 변하게 하는 모핑(morphing), 같은 종류의 다른 모습으로 바뀌게 하는 와핑(warping) 등이 있다.

## 2. 렌더링 및 가시화

컴퓨터에 모델링한 물체나 현상은 뷰잉 파이프라인(viewing pipeline)이라고 불리는 복잡한 과정을 통하여 가시화 된다<sup>[12]</sup>. 세계 좌표계에 위치하는 물체들을 뷰 포트(view port)에 투영을 하고, 윈도우(window)를 벗어나는 물체들은 절단(clipping)되며, 최종적으로 정규좌표계에서 디바이스 드라이버(device driver)에 의해 장치좌표계로 옮겨진다. 현실감 있는 영상을 생성하기 위해 은선/은면 제거(hidden line/surface removal)와 Gouraud 셰이딩, Phong 셰이딩 등이 사용된다.

그리고 총체적(global)으로 음영을 나타내기 위해 광선 추적 모델과 라디오시티(radiosity) 모델들이 사용되고 있다. 또한 위와 같은 음영처리에 의해서는 매끄러운 물체는 잘 표현되지만, 오렌지나 인간의 피부와 같이 거친 피부를 갖는 물체는 표현되는데 어려움이 있으므로, 이를 위해 텍스처 매핑(texture mapping) 기법을 사용하고 있다. 이 외에도 유한 해상도의 영상에서 생기는 계단 현상이나, 화소(pixel)보다 작은 물체가 사라지는 현상, 화소보다 작은 폭을 가지는 긴 물체의 일부가 단절되는 현상과 같은 앨리어싱(aliasing)을 해결하기 위해 안티앨리어싱(antialiasing) 기법 등이 있다.

이와 같이 오늘날 컴퓨터 그래픽스는 크게 표현 범위의 확장과 현실감의 추구로 발전해 왔다. 표현 범위의 확장은 얼마나 다양한 물체를 얼마나 정확하게 표현할 것인가의 문제로서 형상 및 동작의 모델링 및 렌더링이 그 이슈였다. 현실감의 추구는 얼마나 편리하게 컴퓨터를 사용할 것인가의 문제로 사용자 인터페이스가 그 이슈였으며, 궁극적으로는 인간이 컴퓨터에 만들어진 가상 세계에서 현실 세계에서와 같이 행동하고 느끼는 것이 목표였다. 그러면, 앞으로는 어떤 방향으로 컴퓨터 그래픽스가 나아가 할 것인가? 이에 대해 우리는 차세대에서 요구되는 컴퓨터 그래픽스의 소프트웨어, 하드웨어, 그리고 인터넷과의 관계를 조망해 봄으로써 그 답을 찾아보려 한다.

## II. 컴퓨터 그래픽스 하드웨어의 미래

컴퓨터 그래픽스 하드웨어의 미래는 어떠한 것인가? 지금까지 진행되고 있는 움직임을 보면 범용 시스템과 그래픽 전용 시스템에 대한 두 가지 접근이 이루어지고 있다. 범용 CPU들은 아주 빨라지고 있으며 이런 프로세서들이 모여서 모델링, 애니메이션, 렌더링 등과 같은 그래픽 기술들을 수행하고 있다. 반면에 그래픽 전용 하드웨어들은 우수한 기능들을 가지고는 있으나, 실시간 처리의 관점에서만 경제적인 가치를 갖고 있다. 제조업체에

서는 텍스처 맵핑, 안티앨리어싱 그리고 래스터라이징(rasterizing)의 향상에 의한 질 높은 폴리곤(polygon)들을 화면에 생성하는데 노력해 왔다. 그 다음 단계로는 사용자 인터페이스 상에서 3차원 렌더링이 실시간으로 처리될 수 있는 하드웨어의 개발일 것이다. 특히 텍스처 맵핑은 렌더링 하드웨어 개발에 있어서 표면을 인지하는데 도움을 주고 데이터의 가시화에 중요한 역할을 할 것이다. 또한 차세대 컴퓨터 그래픽스 하드웨어는 렌더링 엔진과 픽셀 메모리를 하나의 칩(chip)안에 결합시키므로 해서 상당한 진전을 이루게 될 것으로 전망되고 있다<sup>[21]</sup>. 이러한 결합으로 인해 텍스처 맵핑된 폴리곤들이 하나의 칩에서 병렬(parallel)로 렌더링될 수 있고, 또는 모든 폴리곤들이 픽셀 플레인(pixel plane)안에서 직접 렌더링될 수도 있다.

현재 주종을 이루는 디스플레이 시스템들은 기하학적인 프로세서의 결과물을 래스터 프로세서로 보내는 엔진에 초점을 두고 있다<sup>[22]</sup>. 그러나 지난 15년동안 두 프로세서들을 다양화시키는 작업을 해왔으나, 현재까지 그 기본 구조는 변하지 않고 그대로 남아 있다. 그리고 이런 구조 안에서 디스플레이 하드웨어 디자이너들은 국소(local) 웨이딩, 텍스처 맵핑, 투명효과 등과 같은 여러 가지 이미지 디스플레이를 개발해왔다. 그러나 디스플레이 소프트웨어와 최근의 어플리케이션들은 총체적(global) 그리고 절차적(procedural) 웨이딩 효과를 많이 사용하기 시작했으므로, 이제는 더 이상 이런 전통적인 구조로는 이를 만족시킬 수 없게 되었다.

그래서 웨이딩 효과를 향상시키는 새로운 구조를 제시한 제안들이 나왔다<sup>[23]</sup>. 비록 전통적인 구조를 조금 가지고 있다고 하더라도, 병렬구조(parallelism)로 된 기계에서는 렌더링 효과를 높일 수 있다<sup>[24, 25]</sup>. 그러나 이런 고도의 병렬처리 디스플레이 프로세서들은 웨이딩 효과가 기하학적 처리에 집중되어 있고, 이미지 오퍼레이션(operation)은 효과적으로 다루어지고 있지는 않다. 그래서 컴퓨터 그래픽스 오퍼레이션들의 속도를 증가시키기 위해 범용 컴퓨팅 요소들을 묶는 방법은 감소되고

있는 추세이다.

그러므로 컴퓨터 그래픽스 엔진에 대한 초점은 상호연결(interconnection)에 대한 관심으로 바뀌게 될 것이다. 즉 전문화된 작업이 아니거나 방대한 랜덤 액세스(random access) 대역폭이 필요치 않은 태스크(task)는 병렬·범용 멀티프로세서(general-purpose massively-parallel multiprocessor)들을 이용하는 것이다. 총체적 조명(global illumination), 패스 플래닝(path planning), 모델링, 원격논리 모델링(teleological modeling) 등이 이 범주에 속한다. 컴퓨터 그래픽스 하드웨어가 직접적으로 전반적인 문제를 다룰 것인지, 또는 병렬 범용 멀티프로세서가 다룰 것인지는 아직까지 확실치 않다.

그러면, 장기적으로 볼 때 과연 어떠한 구조에도 적용될 수 있는 컴퓨터 그래픽스 디스플레이 프로세서 구조가 있는지에 대한 근본적인 의문이 생긴다. 가장 좋은 렌더링 방법을 적용한다 하더라도 그래픽 디스플레이 프로세서는 여러 개로 연결된 범용 컴퓨터들에서보다 그래픽 전용으로된 구조에서 훨씬 가속되도록 만들어 질 수 있다. 그럼에도 불구하고 컴퓨터 그래픽스 하드웨어 시스템 디자이너들은 아직도 다음 세대의 새로운 기본 구조를 파악하지 못하고 있다. 그것이 레이(ray)가 될지, 아니면 NURBS가 될지, 솔리드 텍스처(solid texture)가 될지 아무도 모른다. 그러나 궁극적으로는 3차원 입체에 반사되는 광자에 관한 물리 현상이 다른 물리 현상과 유사하므로, 미래의 병렬 구조를 가지는 시뮬레이션 기계에서는 이를 적용하여 쉽게 그래픽 문제를 해결할 수 있을 지도 모른다.

### III. 컴퓨터 그래픽스 소프트웨어의 미래

#### 1. 컴퓨터 그래픽스 라이브러리의 미래

3차원 컴퓨터 그래픽스 어플리케이션의 개발은 대부분 전문가나 매니아들의 몫으로 되어 왔다. 3차원 렌더링과 계산의 요구가 컴퓨터의 수행능력보다 더 커서 일반적인 어플리케이션이나 사용자

인터페이스 개발이 덜 되었기 때문이다. 또한 3차원 기술이 느리게 확산되는 다른 큰 이유는 현재 사용 가능한 소프트웨어 라이브러리가 3차원 프로그래밍에 익숙하지 않은 개발자에게는 사용할 만한 모델을 지원하고 있지 못하다는 점이다. 그러나 RISC기반의 워크스테이션과 PC에서는 이런 3차원 작업이 어느 정도 가능해 보인다. 그리고 하드웨어가 빨라지면 빨라질수록, 소프트웨어의 3차원 어플리케이션 성장의 주요한 요인이 될 것이다.

현재 사용중인 3차원 컴퓨터 그래픽스 소프트웨어 라이브러리는 두 종류가 있다. 하나는 OpenGL이나 Starbase와 같은 하드웨어 드로잉 라이브러리(hardware drawing library)인데, 이것은 프레임 버퍼라고 불리는 하드웨어 디바이스 위에 한 층의 소프트웨어로써 픽셀들과 그래픽 기본 도형들을 그리는 명령어들을 제공하고 있다. 그리고 또 하나는 Phigs+나 HOOPS, Doré와 같은 스트럭처드 드로잉 라이브러리(structured drawing library)가 있는데, 이것은 하드웨어 인터페이스 위에 하나의 레벨로 추상화되고 구조화된 드로잉 명령어들을 제공한다.

3차원 컴퓨터 그래픽스 소프트웨어 툴킷(toolkit)의 다음 세대는 이와 같은 드로잉 기반보다는 객체 기반이 될 것이다. 이런 객체 기반의 툴킷들은 다양한 오퍼레이션 및 편집과 확장이 가능한 3차원 객체의 집합체로 구성될 것이다. 그리고 렌더링은 각 객체가 구현하는 수많은 작업중의 하나가 될 것이다. 이런 3차원 객체들은 서브클래싱(subclassing)과 같은 기능을 통해 프로그래머가 주문하는 형태의 블럭을 쌓게 될 것이다.

미래 3차원 툴킷의 필수적인 요소로 3차원상의 상호작용적인 모델을 들 수 있다. 3차원 객체들은 이벤트(event)들을 받거나 처리하고, 다른 객체에 의해 계층적으로 그룹지어 질 수 있다는 점들에서 2차원 도구들과 비슷하다. 그래서 이런 3차원 객체는 2차원 윈도우 기반에서 상호작용을 위한 슈퍼 셋(super-set)으로도 접근이 가능하다. 그러나 이런 상호작용이 어플리케이션의 일부이고 이는 구현되는 것이 아니라, 3차원 객체의 근본 능력임에 주의해야 한다. 이와 같이 3차원 객체들의 새로

운 클래스는 살아있는 객체를 포함하는 것으로 생성할 수 있다. 예를 들어 이런 객체들 안에 트랙볼(trackball)객체와 마네킹 객체를 포함시켜, 트랙볼로 입력받는 데이터를 마네킹의 관절로 보내어 작용하도록 만들 수 있다. 그러므로 해서 이런 객체들에서의 어플리케이션과 데이터의 경계는 모호해지게 된다.

또한 미래의 객체 지향 툴킷은 현재의 2차원 및 3차원, 윈도우, 사용자 인터페이스 프로그래밍 사이의 구별 또한 모호하게 만들 것이다. 이는 새로운 객체 지향 시스템들이 이것들 모두를 렌더링, 피킹(picking), 이벤트 핸들링, 애니메이션, 그룹핑(grouping), 기하학 계산 및 다른 기본적인 기능을 수행하는 비슷한 객체들로써 다룰 것이기 때문이다. 그러므로 프로그래머가 앞으로 해야 할 일은 컴퓨터 그래픽스와 사용자 인터페이스를 만드는 데 많은 시간을 쏟는 것 보다는 주로 어플리케이션 문제를 해결하는데 초점이 맞추어져야 할 것이다. 게다가, 물리적 성질을 갖는 3차원 객체를 생성하는 것은 훨씬 더 쉬워질 것이고 성행할 것이다. 그래서 어플리케이션들은 간접적인 패널을 이용하는 사용자 인터페이스보다, 실세계의 객체들을 다루는 3차원 사용자 인터페이스를 직접적으로 제공하게 될 것이다.

## 2. 멀티미디어와 컴퓨터 그래픽스 소프트웨어

하드웨어의 고성능 처리 능력과 사용자 인터페이스, 멀티스레드(multithread)등이 지원되는 오퍼레이팅 시스템들과 통합된 미디어들은 많은 컴퓨터 그래픽스 어플리케이션들을 요구하고 있다. 그러나, 요즘의 소프트웨어 프로젝트들에서 현재의 API의 한계가 나타난다. 즉, 멀티미디어를 다룰 수 없고, 타임 크리티컬(time-critical)한 면과 인터랙티브한 컴퓨팅뿐만 아니라 그들의 동기화도 할 수 없다는 한계를 드러냈다<sup>[26]</sup>.

90년대 중반에 이르러 컴퓨터의 보급이 확산되고 컴퓨터의 고성능 처리 능력과 네트워크 능력은 더욱 더 효과적인 사람-기계 인터페이스를 하는 새로운 분산 어플리케이션들을 가능하게 하였다. 그러므로 이런 어플리케이션 사용자들은 컴퓨터

시스템에 실시간으로 접근하고, 타임 크리티컬한 계산과 소리 및 영상 데이터를 다룰 수 있으며, 상호작용을 위한 인간의 동작 감지도 타임 크리티컬하게 할 수 있게 될 것이다. 게다가 어플리케이션들은 네트워크 상에서 공유되는 화이트보드(whiteboard)를 이용하여 원격 협동 작업도 가능하게 할 것이다.

이런 종류의 어플리케이션은 통합된 미디어 구조가 필요하다. 이 구조의 중심은 타임 크리티컬한 계산이 될 것이다. 타임 크리티컬한 계산이란 정확성을 가지고 데드라인(deadline)안에 계산이 끝나야 하는 것을 의미한다. 이러한 계산을 해야 하므로 오퍼레이팅 시스템은 적절하고 정교한 자원(resource)의 스케줄링을 해야 한다. 게다가 자원의 오버로드가 발생할 경우에는 데드라인을 지키기 위해 어느 정도의 결과만을 얻고 계산의 질을 낮추는 것이 타당하기도 하다. 그리고 전문 시스템뿐 아니라, 범용 시스템에서의 멀티태스킹(multitasking)과 네트워크 컴퓨터 환경에서도 타임 크리티컬한 계산은 필수 불가결한 존재이다. 그러므로 이러한 방법을 사용하여 기존의 소프트웨어와의 호환성을 유지하도록 하여야 할 것이다.

통합된 미디어 구조 안에서 3차원 컴퓨터 그래픽스 시스템은 고도의 상호 작용과 치밀한 수학적 모델을 사용하는 중요한 합성 미디어 시스템으로, 치밀한 계산 하에 미디어 스트림(stream)들로 구성되어 있다. 그리고 미디어 스트림의 수신, 압축, 전송, 압축해제를 근간으로 하는 프리샘플(presample)된 미디어 시스템과 비교된다. 즉, 프리샘플된 미디어는 시간이 내재되어 있는 반면에 3차원 컴퓨터 그래픽스 시스템들은 다른 합성 미디어 스트림들간의 동기화(synchronization)와 융합을 얻기 위해 정확한 시간 모델링이 통합되어야 한다.

이와 같이 차세대 컴퓨터 그래픽스를 위해 앞으로 이루어져야 할 중요한 연구들은 다음과 같은 통합 미디어의 구조를 개발하는 것이 될 것이다.

- \* 네트워크상의 타임 크리티컬한 계산
- \* 올바른 형식과 잘 통합된 미디어 전용의

#### API에 대한 규칙들

- \* 서로 다른 미디어 스트림들의 동기화와 융합 및 합성

위와 같이 미래의 3차원 컴퓨터 그래픽스 시스템은 최근의 기본 구조 내에서 동일한 미디어 시스템으로서 역할을 하기 위해서는 시간에 관한 기준을 가장 먼저 세워야만 할 것이다.

### IV. 인터넷과 컴퓨터 그래픽스

인터넷은 빠르게 변화하고 있다. 처음에 인터넷은 ftp, telnet, gopher 등과 같은 문자 위주로 서비스를 하였으나 WWW(World Wide Web)의 등장으로 다양한 멀티미디어 정보와 인터넷으로의 쉬운 접근 방법을 제공하고 있다.

WWW은 HTML(HyperText Markup Language) 문서를 통하여 이용되는데 이것은 정보들을 하이퍼링크(Hyperlink)시켜서 직관적인 정보의 검색을 가능하게 하였다. 여기에서 주목할 것은 2D 이미지가 HTML 문서 속에 포함되어(inline) 같이 제공이 된다는 것이다. 그리하여 사용자들은 기존의 인터넷 방식에 비하여 시각적 정보 검색이 가능하게 되었다. 그러나 HTML으로는 실세계를 표현하는데 한계가 있고 다양한 기술의 발전에도 불구하고 정보의 구성과 그 접근 방식이 2차원에 기반을 두고 있다. 따라서 실세계의 정보를 얻기 위해서는 사용자의 상상력과 지각력이 요구된다.

이러한 문제점을 해결하기 위하여 최근에 인터넷 상에서 3차원 모델의 표준인 VRML(Virtual Reality Modeling Language)이 등장했다. VRML은 많은 사람들의 참여와 관심으로 현재는 다수의 그래픽 관련 회사와 단체로부터 표준으로 인정을 받고 있다. VRML의 등장은 3D 그래픽 기술들을 인터넷에 적용하는 계기를 가져왔으며 이러한 3D 그래픽 기술의 적용은 사용자로 하여금 인터넷에서 몰입감을 가질 수 있게 하고 새로운 정보 검색 방법을 제공하였다.

앞으로 계속 추구해 나아가야 할, 네트워크와 분산환경에 적합한 3차원 컴퓨터 그래픽스의 궁극적인 목표는 완벽한 사이버스페이스(cyberspace)의 구축이다. 그것은 협동적이고 분산되며 상호작용적인 가상 공간으로서, 교육을 받기 위해서나, 새로운 소식, 혹은 오락, 또는 사업상으로 사람들이 모일 수 있는 곳을 뜻한다. 요즘 각광을 받고 있는 자바(Java) 언어는 사이버스페이스의 구축에 중요한 요소들을 지니고 있다. 왜냐하면 자바는 객체지향의 장점을 지니며, 분산 환경에 알맞고 플랫폼에 무관하며 다중스레드를 지원하기 때문이다. 사이버스페이스를 구축하기 위해서 고려되어야 할 사항 중에서 중요한 것들은 다음과 같다<sup>[27]</sup>.

- \* 사이버스페이스를 구축하는 공간 데이터의 수

아무리 거대한 컴퓨터라고 하더라도, 거대한 사이버스페이스를 전부 불러와 렌더링을 하려면 엄청난 부담을 받기 때문에, 가상 공간을 최대한 작게 만들 수밖에 없다. 그러나 실세계를 정교하게 나타내려면 많은 공간 데이터의 수가 필요하다. 따라서 어떤 물체에 가려서 보이지 않는 물체는 제거하여 렌더링을 하지 않아야 한다.

- \* 편재성(ubiquity), 균등성(uniformity), 통일성(unity)의 제공

웹의 근저는 하이퍼스페이스에 있지만, 인간이 탐험할 수 있는 환경을 구축하는데 필요한 세 가지 공간 원리, 곧 편재성, 균등성, 통일성을 제공하려면 하이퍼스페이스를 뛰어넘어야 한다. 이러한 사이버스페이스가 존재하게 되면, 여러 공간을 하나의 연속적인 전체로 연결할 수 있을 것이다. 가령 A의 집이 B의 집과 이웃하고 있는 것은 아니지만 B의 집까지 여행한다면, 중간 공간을 모두 둘러볼 수 있는 것이다. 레이아웃을 통일하면 자신의 위치가 어디쯤 되고 주변에 어떤 정보가 있는지 기억할 수 있다. 이런 조건을 마련하지 못하면 사이버스페이스는 약간 혼란스럽고 연속성이 없다고 여겨질 것이다.

- \* 사이버스페이스의 분산성

어느 정도 정교한 가상 공간은 많은 공간 데이터를 가지고 있는데, 공간 데이터의 재사용 등의 이

유로 다른 웹 사이트에 있는 공간 데이터를 요구할 수 있다. 이러한 사이버스페이스에서는 하나의 세계라는 개념은 더 이상 존재하지 않고, 인터넷 도처에 퍼져 있는 공간 데이터들의 결합이라는 거대한 조합 형태가 된다. 따라서 웹의 여러 곳에 있는 공간 데이터를 결합하여 하나의 사이버스페이스를 만드는 기술이 필요하다.

#### \* 오디오의 지원

웹 페이지에서 배경 화면을 넣듯이 사이버스페이스에서도 배경 오디오를 삽입하여 방향감, 양방향 통신 기술 등을 제공하거나 경험할 수 있는 감각의 폭을 넓힐 수 있다.

#### \* 대화형 사이버스페이스

진정한 사이버스페이스는 대화형이고, 여러 사람이 동시에 참여할 수 있는 환경이다. 그러나 지금까지의 사이버스페이스는 대화 기능이 부족한 반면에 리얼월드는 대화 기능으로 구성되어 있다. 10만 명이나 100만 명이 참여할 수 있는 환경을 구축하려면, 많은 장애물을 극복해야 하지만 5명이나 10명 정도의 사람들이 공유할 수 있는 대화형 공간을 구축하는 것은 그다지 어렵지 않다. 또 지난 몇 년동안 MUD, MOO(텍스트 방식의 사이버스페이스)는 수백여 명의 동시 사용자들에게 대화형 다중 참여 공간을 제공했다.

## V. 결 론

종래에는 과학 기술에 기반을 둔 기술형 부가가치가 가치의 주류를 이루어 왔다. 최근에는 기술형 부가가치와 더불어 예술이나 디자인 등 감성적 부가가치라는 폭 넓은 의미의 가치 창조를 추구하는 시대가 도래하고 있다고 할 수 있다. 기술형 부가가치의 틀을 벗어나 생각하면 컴퓨터 그래픽스는 그 활동 범위의 폭을 더욱 더 넓혀 나갈 것으로 사료된다.

차세대의 컴퓨터 환경에서 요구되는 그래픽스의 요소기술은 어떠한 모습이 되어야 하는가에 관하여 해결되어야 할 과제를 중심으로 살펴보았다. 지

금까지의 컴퓨터 그래픽스는 표현 범위의 확장과 현실감의 추구를 위한 요소 기술로써 모델링, 렌더링, 애니메이션이 그 주요 이슈였다. 그러나 차세대에서 예견되는 컴퓨터 환경은 인터넷과 멀티미디어를 기반으로 하는 오픈 시스템 환경에서 운용되어 질 것이며 컴퓨터 그래픽스 또한 이러한 환경을 고려한 발전 방향이 설정되어야 할 것이다. 이러한 관점에서 본고에서는 차세대에서 해결하여야 할 컴퓨터 그래픽스 요소 기술에 대해 컴퓨터 그래픽스 소프트웨어 시스템, 하드웨어 그리고 인터넷과 3차원 컴퓨터 그래픽스와의 관계들을 논함으로써 그 답을 찾아보았다.

#### 참 고 문 헌

- [1] J. Foley et al., "Computer Graphics : Principle and Practice," 2nd ED. Addison-Wesley, 1990.
- [2] J. Boyse and J. Gilchrist, "GMSolid : Interactive modeling for design and analysis of solids," CG&A, 2(2) : 27-40, 1982.
- [3] I. Braid, "Notes on a geometric modeler," Technical Report CAD Group Document 101, Computer Laboratory, Univ. of Cambridge, 1980.
- [4] A. Requicha. and H. Voelcker, "Solid modeling : Current status and research directions," CG&A, 3(10) : 25-37, 1983.
- [5] H. Voelcker et al., "The padl-1.0/2 system for defining displaying solid objects," SIGGRAPH "78, page 257, 1978.
- [6] W. Reeves, "Inbetweening for computer animation utilizing moving point constraints," SIGGRAPH "81, pages 263-269, 1981.
- [7] A. Barr and R. Barzel, "A modeling system based on dynamic constraints," SIGGRAPH, "88, pages 179-188, 1988.

- [8] J. Platt and A. Barr, "Constraint methods for flexible models," SIGGRAPH '88, pages 279-288, 1988.
- [9] J. Hahn, "Realistic animations of rigid bodies," SIGGRAPH '88, pages 299-308, 1988.
- [10] B. Mandelbrot, "Fractals : Form, Chance and Dimension," W. H. Freeman, 1977.
- [11] H. Peitgen and P. Richter, "The Beauty of Fractals : Images of Complex Dynamical Systems," springer-Verlag, 1986.
- [12] R. Voss, "Fractals in nature : Characterization, measurement, and simulation," In Course Notes 15 for SIGGRAPH "87, 1987.
- [13] T. Loke et al., "Rendering fireworks displays," CG & A, 12(3) : 33-43, 1992.
- [14] W. Reeves, "Particle systems—a technique for modeling a class fuzzy objects," SIGGRAPH "83, pages 359-376, 1983.
- [15] K. Sims, "Particle animations and rendering using data parallel computation," SIGGRAPH "90, pages 405-413, 1990.
- [16] Adole, "The Display Postscript System-Reference Manual," Adole Systems Incorporated, 1991
- [17] D. Breen, D. House, and P. Getto, "A particle-based computational model of cloth dropping behavior," The Visual Computer, 7 : 113-134, 1991.
- [18] M. Carignan et al., "Dressing animated synthetic actors with complex deformable cloths," SIGGRAPH "92, pages 99-104, 1992.
- [19] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," IEEE Trans. Pattern Analysis and Machine Intelligence, 8 : 413-424, 1986.
- [20] D. Terzopoulos et al, "Elastically deformable models," SIGGRAPH "87, pages 205-214, 1987.
- [21] Douglas Voorhies, "Beyond Gouraud-Shaded Polygons... Where Will Graphics Hardware Go Next?," SIGGRAPH '92 Chicago, July 26-31, 1992, pages 412-413.
- [22] Reghbati, H. K. and A. Lee, "Tutorial : Computer Graphics Hardware—Image Generation and Display," IEEE Computer Society Press #753, 1988.
- [23] Clark, James H., "Roots and Branches of 3-D," Byte, v.17, no.5, May 1992, pp.153-164.
- [24] Fuchs, Henry, et al., "Pixel-Planes 5 : A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories," Proceedings of SIGGRAPH '89, pp. 79-88, July 1989.
- [25] Potmesil, Michael, and Eric M. Hoffert, "The Pixel Machine : A Parallel Image Computer," Proceedings of SIGGRAPH "89, pp.69-78, July 1989
- [26] Andries van Dam, Graphics Software Architecture for the Future SIGGRAPH '92 Chicago, pp. 389-390, July 26-31, 1992.
- [27] Mark Pesce, "VRML-Browsing Cyberspace," New Riders Publishing 1995, 6.



## 저자 소개



金昌憲

1957年 5月 1日生

1979年 2月 고려대학교 학사

1993年 2月 Tsukuba 대학교 전자·정보공학 박사

1981年 9月~1995年 2月 한국과학기술연구원 책임연구원

1989年 4月~1990年 3月 Tokyo Institute of Technology 객원연구원

1993年 9月~1995年 2月 충남대학교 컴퓨터공학과 겸임교수

1995年 3月~현재 고려대학교 컴퓨터학과 조교수

주관심 분야: 컴퓨터 그래픽스, 가상현실