

**3D Animation 교육에 대한 개념적 접근 방식에 대한 연구**  
A Study on the educational method of the conceptual approach of 3D Animation

최 성 원(Choi, Sung-Won)

동명정보대학교

1. 서 론

- 1-1. 연구목적
- 1-2. 연구내용 및 방법

2. 현실세계와 3D Software와의 관계

- 2-1. 차원의 의미 - 대상들의 부피성
- 2-2. 대상과 그 속성 - Modeling과 Mapping(Texturing)
- 2-3. 빛 - Light
- 2-4. 눈(眼) - Camera
- 2-5. 운동 - Animation
- 2-6. 각인 - Rendering

3. 3D Software의 논리성

4. 결 론

(要約)

현재 많은 사람들이 3D Software를 사용하고, 배우고 있으나, 이를 도구의 개념으로 사용하기보다는 이미지를 창출하는 도구로 사용하고 있는 실정이다. 이러한 경우 3D Software의 변화는 사용자에게 심각한 문제로 대두되어질 것이다. 이는 대부분의 경우에 있어서 3D 전반에 대한 이해 부족 그리고 사용자가 3D Software에 대하여 일종의 환상을 가지고 그것의 기능에만 열중한 결과라고 할 수 있겠다. 이는 대부분의 대학의 교육에서도 상아탑의 본질이 3D Software의 기능만을 가르치는 기능인 양성소의 장으로 변질 되어가고 있는 실정이다. 따라서 본 연구는 이러한 폐단을 막기 위해서 가상세계라고 불리는 3D를 현실세계와의 비교에 의한 개념 즉 어떻게 인간이 세상을 刻印하고 있는가를 토대로 일반적인 3D Software의 구성도를 설명 하고자 한다. 이러한 기본적인 설명과 이해가 뒷바침된 후, 3D Software에 대한 기능 설명이 점진적으로 구체화되어진다면, 3D Software에 대한 사용자들의 이해가 더욱 더 빨라질 것이며, 처음 대하는 사용자라고 할지라도 그리 어렵게 접근하지

않을 것이며, 대학의 교육은 더 이상 3D Software만을 가르치는 교육의 장이 아니라 원래의 모습으로 그 기능을 다할 수 있을 것이다.

(Abstract)

Nowdays many people using and learning 3D Software. However, like artists drawing by brush, that most of designers misunderstood that 3D Software is cover everything as a tool. This sort of situation, change of 3D Software will come to the front very seriously. It can be result about lack of understanding 3D in general and some kind of illusion about3D Software. These were changing like trining school that was teaching only function for technique. Therefore, to prevent above things, this research paper will be based on a conception, that compare with 3D which called cyber space and the real world. That is explanation of construction about 3D Software generally, that will basis how does people think about the world. If function of 3D Software will take concrete shape gradually, which is after general description and understanding and the users will be quick to understand, no need to difficult of access for beginners. In addition, higher education ay university is no longer training center. Lastly, it will be able to capability as an essentially figure.Nowdays many people using and learning 3D Software. However, like artists drawing by brush, that most of designers misunderstood that 3D Software is cover everything as a tool. This sort of situation, change of 3D Software will come to the front very seriously. It can be result about lack of understanding 3D in general and some kind of illusion about3D Software. These were changing like trining school that was teaching only function for technique. Therefore, to prevent above things, this research paper will be based on a conception, that compare with 3D which called cyber space and the real world. That is explanation of construction about 3D Software generally, that will basis how does people think about the world. If function of 3D Software will take concrete shape gradually, which is after general description and understanding and the users will be quick to understand, no need to difficult of access for beginners. In addition, higher education ay university is no longer training center. Lastly, it will be able to capability as an essentially figure.

(Keyword)

3D Software, Realism

## 1. 서론

### 1-1. 연구목적

만일 어떤 화가에게 그가 사용하던 붓 대신에 다른 붓을 쥐어 주었을 때, 자신의 세계를 표현하지 못한다면 당신은 그를 과연 화가라고 할 것인가? 그와 같은 맥락에서 Computer Graphist 그것도 3D Computer Graphist에게 비록 그가 사용하던 Software이지만 완전히 인터페이스가 바뀐 또는 다른 종류의 3D Software를 주었을 경우, 자신의 세계를 표현하지 못한다면 당신은 과연 그를 3D Computer Graphist라고 말할 수 있겠는가? 이는 현재 3D Software를 공부하려는 사람들뿐만 아니라 실무에서 종사하는 3D Computer Graphist들에게도 가장 심각하게 대두되어지는 문제이다. 이러한 문제들을 자세히 살펴보면, 사용자는 3D Software를 하나의 도구로서가 아닌 해결물로 생각함과 더불어 3D 전반에 대한 이해 부족, 그리고 3D에 대하여 사용자가 일종의 환상을 가지고 그것의 기능에만 열중한 결과라고 할 수 있겠다. 이러한 문제점 때문에 현재 많은 비판가들에게 과연 3D Software는 '작가의 세계를 표현하기 위한 수단으로서의 Software인가 아니면 Software에 의해서 결정되어진 표현의 세계인가?' 하는 비판의 대상이 되고 있다.

이러한 현상은 현재 Software 시장이 일반적으로 3년을 주기로 완전한 변화가 이루어지고 있지만, 앞으로는 이러한 주기가 더욱 더 짧아질 것으로 예상되어지기 때문에 더욱 더 심각한 문제로 대두되어질 것으로 보여진다. 따라서 본 연구는 이러한 문제점을 해소하기 위해서는 대학 내에서의 3D Software 교육에서부터 그러한 문제점을 주시하고, 올바른 3D Software 교육을 시작해야 함을 인식하고, 현 대학교육에 있어서 어떻게 학생들에게 3D Software를 그 근본개념에 따른 도구개념으로 이해시키고 교육시킬 수 있는가에 그 연구 목적이 있다. 이러한 본 연구의 목적은 결과론적으로는 3D Software를 한번도 대해보본 적이 없는, 또는 자신이 사용하던 3D Software의 인터페이스가 완전히 변화하였을 경우, 학생들이 어떻게 가장 빠르고, 효율적으로 습득할 수 있게 하는가에 대한 연구이기도 하다.

### 1-2. 연구내용 및 방법

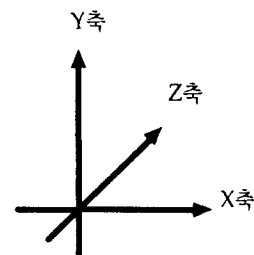
본 연구는 어떤 일정한 Software를 기준으로 설명해 나가는 것이 아니라, Software일반에 대한 설명이기 때문에 3D Software의 가장 근본적인 내용을 주제로 삼고 있다. 즉 3D Software는 우리의 세상을 모방하여 영화와 같이 Computer에 재건하기 위해서 만들어진 Software인 만큼 연구의 내용은 모든 Software의 가장 기초에 깔려 있는 '인간이라는 종으로서 세상이 어떻게 우리의 눈에 각인 되어지는가에 대한 논리적 사고'와 그에 해당하는 3D Software의 대개념들과의 상관관계를 그 내용으로 삼겠다. 그 방법론으로는 우리의 눈에 각인 되어진 세상을 프로그래머들이 철학자의 세계에 대한 인간의 인식 방법론을 Software화 하여 사용자로 하여금 자신의 세계를 표현하도록 제작하였기 때문에 본 장에서는 프로그래머의

입장에서 이해해야 한다는 것이 본 연구의 방법론이 되겠다. 그러나 여기서 프로그래머들이 만든 알고리즘을 설명하는 것이 아니라, 인간의 일반적인 사고 논리를 일반 사용자로 하여금 쉽게 접근하게 하기 위하여 프로그래머들이 어떤 방법론을 어떻게 사용했는가에 대한 문제를 축으로 본 연구 방법이 진행되어질 것이다.

## 2. 현실세계와 3D Software와의 관계

### 2-1. 차원의 의미 - 대상들의 부피성

과연 3차원의 의미는 무엇일까? 이러한 차원은 시각적 대상이 아니기 때문에 그 의미를 알기 위해서는 해당되는 차원 내에 존재하는 대상들의 공통점을 찾아내어 유추할 수밖에 달리 도리가 없다. 그럼 3차원 공간 내에 존재하는 모든 대상들의 공통점은 과연 무엇일까? 우리의 눈에 보여지는 모든 대상들 즉 種 개념으로서 대상일반의 공통점을 가만히 생각해보면, 모든 대상은 부피를 가지고 있으며, 이러한 부피 때문에 우리 눈에 모든 대상이 각인 되어지는 것이다. 이렇게 부피를 가지고 있는 대상은 그 자체로서 X, Y, Z 축을 기본으로 하고 있기 때문에 두 축으로 이루어진 면과 하나의 축만을 갖는 선 그리고 선을 이루고 있는 점의 세단계를 모두 내포하고 있다. 그러나 이러한 점, 선, 면은 부피를 이루는 필수 조건이지만 이들 자체가 3D 공간 내에서 우리의 눈에 보여지는 것은 아니다. 따라서 우리는 현실세계의 3차원 대상의 부피가 면, 선 그리고 점을 포함하고 있으며 이들은 각각 2차원, 1차원, 0차원을 구성하는 특징이기 때문에 3차원은 그 이하의 차원을 포함함과 더불어 대상의 부피를 가능하게 하는 특징을 갖고 있음을 알 수 있다. 이러한 특징을 갖는 현실 공간세계를 그대로 옮겨 놓은 컴퓨터 내의 3차원 가상 세계는 수학(특히 기하학)에 기초하여 만들어졌기 때문에, 역시 마찬가지로 그 이하의 모든 차원과 그에 해당하는 대상들을 포함하며, 이를 기초로 이루어지고, 마지막으로 현실세계와 같이 부피에 의해서 완성되어진다. 이러한 3차원의 구성도는 XYZ라는 좌표에 의해서 하기와 같이 표현된다.



### 2-2. 대상성과 그 속성 - Modeling과 Mapping (Texturing)

미국 해병에서 컴퓨터의 3차원 공간을 우연하게 발견한 이후로 많은 프로그래머들은 이러한 3차원의 공간에 현실 세계와 같은 Realism을 가진 세상을 구축할 수 없을가 하는 호기심을

갖게 되었다. 그러한 결과로 프로그래머들은 철학을 기반으로 모든 대상들이 갖고 있는 공통점에 대하여 연구하기 시작하였으며, 이를 기초로 만들어진 Software들이 바로 우리가 현재 사용하고 있는 3D Studio Max, Softimage, Maya, Light Wave, Shade, Strata등과 같은 것들이다. 이러한 모든 Software들은 사용자에게 그의 세계를 표현해줄 수 있는 場을 만들어 준 것일 뿐, 어떤 구체적인 것도 Software 자체가 만들어 낼 수는 없다. 이는 자칫 잘못 생각하면 Software 사용자가 프로그래머들에 종속되어 있는 듯한 느낌을 줄 수 있지만, 사용자에게 의해서 Software는 비로소 Software일 수 있기 때문에 어떠한 경우에도 열등관계를 이루지 않는다. 그러나 만일 Visual 문제가 강조되어진다면, 그러한 가상의 세계를 만들어 내는 Software 사용자의 역할이 오히려 강조되어질 수 있다. 그럼 다시 일반론으로 돌아와서, 모든 Software는 비록 제조회사가 다르기는 하지만 부피를 만드는데 있어서 상기에 언급한 바와 같이 기하학의 기초를 이루는 점에서 출발하여 점들의 결과물인 선 그리고 선들의 집합체인 면을 만들며, 그 결과로서 부피를 가진 대상을 만드는 공통점을 가지고 있다. 따라서 사용자들은 3D에서 만들어지는 대상은 점에 의해서 표현되어지며, 이러한 점들의 위치는 대상의 구체적 형태를 결정 지우는 것이기에 3D 대상의 가장 기본이 되는 點 역할의 중요성을 잊지 말아야 할 것이다. 이렇게 점, 선, 면을 기초로 하나의 대상이 다른 대상과의 차별성을 이루는 형태의 문제는 해결되었지만 진정한 차별성의 문제는 아직 해결되어지지 않은 채 여전히 남아 있다. 그럼 나와 너를 구별되어지게 하는 개념은 과연 무엇일까? 즉 대상을 대상으로서 독자적으로 존재하게끔 하는 대상임은 무엇일까? 그것은 다른 아닌 대상의 속성인 것이다. 즉 대상이 어떤 형태, 어떤 색상, 그리고 어떤 재질을 갖고 있는가에 따라 하나의 대상은 다른 대상과 진정으로 구별되어진다. 이는 대상의 개별성을 근거 지우는 개별자의 특성인 것이다. 이로서 대상은 부피와 속성으로서 개별자임이 정의되어지고, 이러한 개별자들은 다음과 같이 분류되어진다.

\* 가시적 대상

- 고정적 대상 : 시간의 흐름과 더불어 생성, 소멸 등은 가능 하지만 타자에 의한 자발적 운동 외에는 가능하지 않은 것.

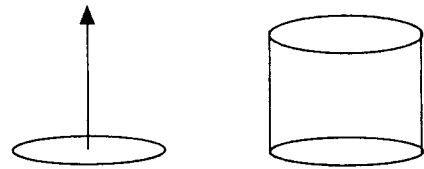
- 유동적 대상 : 고정적 대상과 마찬가지로 시간의 흐름과 더불어 생성, 소멸 등이 가능하며 자발적 운동이 가능하다.

\* 비가시적 대상 - 우리의 눈에는 보이지 않으나 자연에 존재하는 보편적 법칙(예 : 중력, 자력 등)

이상에서 살펴 본 바와 같이 대상은 부피와 속성으로 정의어진다. 이러한 것을 프로그래머들은 3D Software에서 대상의 부피와 자신만의 고유한 모습을 재현시킬 수 있는 場으로서 Modeling 분야를, 그리고 속성인 색상과 재질을 만들 수 있는 場으로서 Mapping 또는 Texturing (비가시적 대상의 경우에는 예외적이다)이라는 분야를 만들어 사용자로 하여금

Software를 이용하여 현실의 Realism을 특징으로 갖는 가상의 대상들을 만들 수 있도록 설계하였다. 그럼 이러한 대상들을 만들 수 있는 가장 기본적인 방법론들은 과연 무엇일까? 여기서 그러한 방법론을 일일이 언급하는 것은 불가능한 일이지만 그러한 대상을 만드는 가장 근본적인 방법론에 대해서 살펴보겠다. 우선 주위의 대상들을 살펴보면 좌우대칭(예 : 컵)이 되는 것들 그리고 하나의 단면이 어떤 곡선을 일정하게 따라 가는 대상(예 : 난간의 손잡이)들 완전히 불규칙한 대상들(예 : 자연적인 것)을 발견 할 수 있다. 이러한 대상들을 만들기 위해서 사용자는 Software에서 제공한 점으로부터 출발하여 선과 면을 부피화 시키는 다음과 같은 방법이 있다.

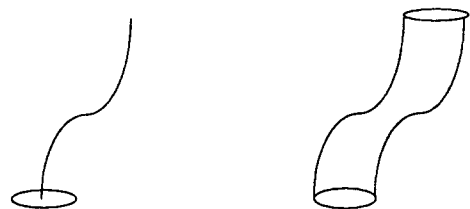
- Elevation : 하나의 축을 중심으로 하나의 면을 그대로 상승 시킴.



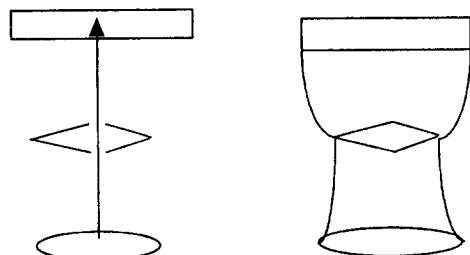
- Revolution : 하나의 축을 중심으로 대상의 단면 곡선을 사용자가 원하는 값만큼 회전시킴.



- Extrusion : 대상의 단면을 X, Y, Z 또는 사용자가 만들어 놓은 곡선을 따라 단면을 부피화 시킴.



- Skinning : 형태와는 상관없이 같은 점을 가진 선들 또는 면들이 조합하여 부피를 가짐.



이상의 기능들이 모든 Software의 Modeling에 포함되어 있고 이를 기초로 대상을 제작하며, 그 외의 것들은 대부분의 경우에 있어서 이들 기능들의 응용이라고 할 수 있겠다. 그러나 여기서 잊지 말아야 할 것은 모든 Software가 Edit의 기능을 갖고 있듯이 3D Software역시 Edit의 기능을 갖고 있다. 그러나 앞에서 언급한 바와 같이 부피를 갖는 대상들의 기본적인 요소는 점들이기 때문에 대부분의 Edit 기능들은 점(들)을 수정하는 것에 초점이 맞추어져 있다. 이러한 점들을 제어할 수 있는 Edit의 기능은 어느 Software에나 존재한다. 이러한 Modeling과 Editing으로 비록 대상을 제작하였을지라도 많은 경우에 있어서 하나의 구체적 Model은 여러 개의 대상 - 테이블의 경우는 탁자와 4개의 다리 - 으로 이루어져 있기 때문에 대상의 위치, 크기 등을 정확히 맞추기 위해서 대상들을 이동시키고(Translate), 회전시키고(Rotate), 확대, 축소(Scale)하는 기능들이 기본적으로 3D Software에 존재한다. 그럼 이렇게 대상들이 자신만의 부피를 가졌다는 가정 하에서, 대상의 재질 또는 색상은 3D Software에서 어떻게 가능한 것일까? 바로 이러한 부분을 위해서 Software Programmer들이 만들어 낸 것이 Mapping 또는 Texturing이라고 하는 부분이다. 이는 인위적으로 사용자가 만든 대상에 재질, 이미지 또는 색상을 부여하는 것이므로 Modeling의 경우와 마찬가지로 다분히 인위적이다. 이렇게 사용자의 임의대로 할 수 있는 만큼, 현실에서 불가능한 것들 또한 가능하다. 여기서 가장 기본이 되는 것은 대상에 색상을 입히는 경우인데, 이 경우에는 대상의 Ambient(분위기), Specular(빛의 반사), Diffuse(빛의 확산)의 표현을 RGB(red, green, blue)의 세 가지 색상 모드로 표현한다. 그리고 만일 대상이 어떤 특수한 형태의 재질 또는 이미지 등을 가지고 있다면, 그에 해당하는 이미지들을 스캐닝한 후 이를 대상에 적용시켜 그만의 이미지와 재질을 적용시킨다. 대상 전체에 어떤 색상을 적용하는 것은 RGB 모드의 조절로 그리 어렵지 않게 수행할 수 있으나, 어떤 재질 또는 이미지를 적절하게 적용시키기 위해서는 적용 하고자 하는 이미지와 대상의 비율 그리고 위치 등의 상관관계를 정확하게 이해 해야한다. 이는 대상의 Realism을 구체화시키는 것이며, 현재는 이를 통하여 많은 특수효과를 재현 할 수 있도록 제작되었다.

이상과 같은 Modeling과 Mapping 또는 Texturing을 도식화하면 다음과 같이 나타낼 수 있다.

	기본요소	기본적인 수정기능	기타
Modeling	Elevation, Revolution, Extrusion, Skinning	점들을 중심으로 수정하는 것을 기본	정확한 대상을 만들기 위해서는 점들의 표현이 중요하다
Mapping 또는 Texturing	RGB mode와 스캐닝한 또는 자신이 만든 이미지	적용할 이미지를 수정함으로써 가능하다.	재질의 표현은 대상에 대한 이미지의 비율과 위치

### 2-3. 빛 - Light

그럼 다시 현실세계로 돌아와 대상이 어떻게 인간의 눈에 각

인 되는지 그 과정을 살펴보자. 부피와 속성을 지닌 대상이 존재한다고 가정할 때, 과연 이러한 두 가지의 조건만으로 모든 것을 만족시킨다고 말할 수 있을까? 이는 현실세계의 빛 - 그것이 자연적인 또는 인위적인 빛이든 - 의 존재 없이는 어떠한 대상도 우리의 눈에 각인 되어질 수 없다. 따라서 대상의 모습은 부피와 속성을 지닌 대상에 투사되어진 빛과의 접촉점에 의해서 우리의 눈에 들어오는 대상의 모습이다. 따라서 빛은 대상의 모습을 우리의 눈에 비추게 하는 필수조건이다. 이러한 과정을 통해 우리의 눈에 각인 되는 대상의 모습이기에 아이러니컬하게 우리는 진정한 대상 자체의 모습을 알 수 없다는 결론이 나타난다. 이러한 현실세계의 상황을 이해한다면 얼마나 빛의 역할이 중요한지 알 수 있다. 3D Software에서는 이러한 현실의 빛을 대신하는 Light가 존재한다. 이는 현실의 자연적 그리고 인위적인 빛 모두를 포함하고 있다. 그러나 비록 현실의 빛을 모방하여 만들었지만 프로그램적으로 완전히 똑같이 작용하지 않는다는 것을 명심해야 한다. 3D Software내의 빛의 종류는 자연 빛, 인위적인 빛을 모방한 Sun과 Spotlight를 기본으로 하고 있으며, 그에 대한 색상은 빛의 삼원색인 RGB로 결정하게끔 되어 있다. 여기서 Sun은 현실세계의 태양과 같이 전체를 비추는 기능을 갖고 있고, Spotlight는 인위적인 빛으로서 영화의 그것과 같은 것이며, 대개의 경우 부분 조명으로 사용된다. 이는 Target과 Spotlight 자체로 구분되어지며, 대상에 대한 Spotlight의 Focal 각도에 따라서 빛이 비추는 범위가 결정되어진다. 이러한 Spotlight는 Hotspot과 Falloff로 구분되어지는데, Hotspot은 빛이 중점적으로 밝히는 부분이며, Falloff는 그 중심으로 부터 점점 빛이 어두워지는 부분을 말한다. 즉 빛이 Gradient 되어 Spotlight의 빛이 점차 상설되어지는 부분을 말한다. 이러한 빛의 두 종류는 Software에 출현한 다른 빛들의 기본이 되어진다고 할 수 있으며, 이러한 빛은 현재에 와서 단순히 어느 범위를 비추는 조명 효과 외에도 이를 이용한 다양한 효과를 얻을 수 있다. 이상에서 알 수 있듯이 빛의 적절한 사용은 사용자가 만든 대상과 그 속성을 우리에게 온전히 전해줄 수 있는 중요한 요소인 것이다.

### 2-4. 눈(眼) - Camera

대상과 빛이 존재함으로써 비로소 대상은 우리의 눈에 각인 되어질 준비가 되어 있다. 그러나 우리의 눈이 존재하지 않는다면 대상은 대상으로서 우리에게 인식되어지지 못한다. 왜냐하면 대상이란 우리에게 대한 대상일 뿐 그 자체로서는 아무런 의미도 갖고 있지 않기 때문이다. 따라서 대상이 대상으로서 존재할 수 있는 조건은 우리의 눈이 필수조건이며, 이로서 우리는 부피를 가진 대상의 사면을 살펴볼 수가 있는 것이다. 즉 우리가 현실에서 대상을 살펴 볼 수 있는 근거는 우리가 자발적으로 고개운동을 하거나, 신체적 운동을 통해서 눈으로 대상의 형태를 보는 것이다. 대상을 보는 우리 눈의 각도는 매우 다양하며, 대상의 포착력은 언제나 시간적 공백 없이 그 즉시 이루어진다. 그러나 이러한 대상에 대한 집중력은 우리의 관심에 의해서 이루어지기 때문에 시각의 집중력은 우리의 주관에 기초한다. 이는 영화에서 감독의 철저한 계획과 그의 주관성에 기인하는 Camera와 같으며, 이는 3D Software에서

도 역시 Camera로 표현된다. 즉 Camera는 영화의 그것과 같고 또한 현실세계에서 인간의 눈에 비유되어진다. 따라서 3D Software에서의 Camera는 사용자의 치밀한 계획과 더불어 그의 주관에 의해서 포착되어진다고 말할 수 있다. 3차원 공간 내에서의 Camera는 사용자의 정의에 의해서 영화에서도 불가능한 어떠한 이동, 회전도 가능하며, 사용자가 원하는 어느 부분이라도 쉽게 포착해 낼 수가 있지만, 쉽게 제어되는 만큼 사용자의 신중함과 치밀한 계획에 의해서 이루어져야 한다. 이는 기능적으로는 Light와 마찬가지로 Target과 Camera자체로 이루어지며, 그와 더불어 Focal에 의해서 Camera 앵글 범위를 정하도록 되어 있다.

## 2-5. 운동 - Animation

이상의 모든 것은 컴퓨터내의 가상 공간 내에서만 이루어질 수 있는 것이다. 여기에 시간의 의미가 개입하면서 운동이 시작된다. 즉 운동은 시간이 전제되어야만 가능한 것이다. 이를 위해서 프로그래머들은 시간의 의미를 Frame으로서 사용자에게 제공해 주고 있다. 이로서 가상의 공간에서 한 걸음 더 나아가 가상의 시공간으로 의미가 확장되어진다. 그럼 우리가 상기에서 언급한 대상, 빛 그리고 카메라 중 운동이 불가능한 것이 있을까? 현실의 빛과 우리의 눈은 운동을 하기 때문에 현실을 모방한 빛과 카메라는 운동이 가능하다. 그럼 대상은 어떠한가? 상기에서 고정적인 대상이 있다고 말했다. 그러나 이것조차도 타자에 의해서 운동 가능하고, 또한 시간의 흐름에 따라 마모 내지는 소멸 가능한 대상이기 때문에, 그것이 비록 고정적 대상이라고 할지라도 그리고 대상의 속성이라고 할지라도 운동 가능하다. 이로서 우리는 시간의 흐름에 따라 변하지 않거나, 운동이 되어지지 않는 대상은 없다고 결론 지을 수 있다. 따라서 공간을 대상이 존재할 수 있는 근거라고 한다면, 시간은 대상이 운동할 수 있는 근거라고 할 수 있다. 이러한 운동의 종류를 살펴보면, 다음과 같은 공통점을 가지고 있음을 알 수 있다. 즉 이동(Translation), 회전(Rotation) 그리고 확대와 축소(Scaling)의 Animation이 가능하며, 이는 대상, Light 그리고 Camera 모두에 해당한다.(현재는 Mapping과 Texturing의 Animation도 가능함) 여기서 우리가 주지해야할 사항은 3D의 모든 대상은 Hierarchy - 이는 Modeling에서 사용자가 하나의 점을 찍기 시작할 때부터 Computer내에서 정의되어진다 - 를 갖는데 이것은 Computer내에서의 대상의 중심이며, 대상은 이를 중심으로 이동, 회전, 확대, 축소한다. 그러나 대상이 항상 자신의 중앙 부분에서 운동되어지지 않는 만큼 - 예를 들어 출입문의 회전운동은 출입문의 중앙부분에서 이루어지는 것이 아니라 출입문의 좌측 또는 우측부분이 중심이 되어 회전한다 - 사용자에게 의해서 Hierarchy의 이동은 가능하다. 이러한 Hierarchy는 대상의 중심이며, 대상을 움직이게 하는 근본이기 때문에 모든 3D 대상은 이 Hierarchy에 종속되어 있다. 그 뿐만 아니라, 이것의 중요성은 Animation에 있어서 대부분의 대상이 종속관계 - 예를 들어 인간 신체의 중심은 몸통이며 여기에 팔, 다리, 얼굴이 종속되어 몸통을 따라가게 되어 있다. 그렇지 않을 경우 이러한 Animation은 아마도 불가능할 것이다. - 를 가지고 있기 때문에 이 종속관계를 유지시키는 근본이기도 하다. 이러

한 종속관계는 3D Software에서는 비록 시각적으로는 대상이 대상에 종속되는 것처럼 보여지지만 실제로는 Hierarchy에 대한 Hierarchy의 종속인 것이다. 따라서 일반적인 Animation의 기초는 대상의 중심을 이루는, 그리고 사용자에게 의해서 이동 가능한 Hierarchy인 것이다. 이상에서 언급한 Animation은 대상에만 국한되어진 Animation이 아니라 Light(및 Target), Camera(및 Target)에도 해당한다. 그럼 대상의 변형에 대해서는 어떠한가? 이것을 위해서 Morphing 또는 Interpolation이라는 Animation이 존재한다. 그러나 이는 현실에서와 같이 하나의 대상자체가 변형되는 것이 아니라, 같은 수의 점을 가진 두 대상이 사용자의 시간 정의에 따라 하나의 대상에서 다른 대상으로 변하는 것이다. 즉 같은 수의 점을 가진 어린이와 할아버지의 얼굴이 있다면, 사용자의 시간적 정의에 따라 어린이의 얼굴이 할아버지의 얼굴이 변해 가는 것이다. 이는 3D Animation의 白眉라고 할 수 있으나, 변화하는 과정을 사용자가 잘 알고 있어야 효과적인 Morphing 또는 Interpolation이 가능하다. 이상에서 말한 것과 같이 Animation은 시간, 정확히 말하자면 Frame과 관계되며, 이러한 시간은 속도와 더불어 Animation의 힘과 관계를 갖기 때문에 시간의 완급조절은 Animation의 생명이라고 할 수 있겠다.

## 2-6. 각인 - Rendering

이제 우리는 대상이 우리의 눈에 각인 될 수 있고, Animation이 가능한 모든 조건을 갖추었다. 부피와 속성을 가진 대상, 이것을 비추어줌과 동시에 대상을 우리의 눈에 반사 해주는 빛, 그리고 그러한 대상을 비로소 대상이게끔 하는 우리의 눈, 이 세 가지가 현실에서 우리가 대상을 각인 하는 조건이며, 이를 모방하여 3D Software에서는 Modeling, Mapping 또는 Texturing, Light, Camera 그리고 Animation이라는 부분으로 구분해 놓았다. 그러나 문제는 현실의 세계에서는 우리의 눈이 대상을 볼 경우, 모든 대상이 실시간 적으로 아무런 지장 없이 각인 되어지는 반면, 컴퓨터에서는 현실과 같이 실 시간적 각인이 불가능하다는 것이다. 따라서 이러한 것을 보완하기 위해서 만들어 놓은 부분이 바로 Rendering이다. 따라서 Rendering은 실세계에 비유를 한다면 비로소 대상이 부피와 속성을 가진 대상으로서, 바로 우리의 눈에 각인 되어 우리가 대상을 대상으로 인식하는 과정인 것이다. 이러한 Rendering은 인간의 시각적 착각에 의해서 보여지는 셀 애니메이션과 같은 시스템을 쓰고 있다. 즉 시간에 따른 각각의 Frame이 정지된 이미지로 나타나며, 이러한 일련의 이미지들의 연속성이 우리들의 눈에 움직이는 것처럼 보여지는 것이다. 이는 TV를 예로 든다면, NTSC 방식을 선택한 한국, 일본, 미국 등과 같은 나라들은 초당 30Frame(유럽은 Pal/Secam방식으로 초당 24 Frame), 즉 1초의 Animation을 위하여 30개의 정지된 이미지를 Computer가 계산한다. 이는 3D의 마지막 작업이며, 이러한 Rendering이 끝나면 사용자는 이러한 일련의 정지된 연속적 이미지를 시청자에게 보여주는 것이다. 따라서 사용자는 비록 3차원 공간에서 작업을 수행 하지만, 결과적으로 보여지는 것은 Rendering에 의한 결과물로서 2차원 평면으로 보여지는 것이다. 따라서 3D영화라는 것은 3D로 작업한 영화인 것이다.

이상과 같이 현실세계에서 인간의 눈에 각인 되는 방식을 3D Software와 비교한 내용을 표로 만들면 다음과 같다.

	기본기능	주지사항	Translation, Rotation, Scale
Modeling	Elevation:단면 상층 Revolution : 단면 회전 Extrusion : 단면을 선에 따라 상층 시킴 Skinning : 단면들을 조합하여 부피를 가짐	기본요소는 점들이기 때문에 이러한 점들을 제어할 수 있는 Edit 기능이 존재한다. 그외의 기능들 이를 응용, 발전 시킨 기능이다.	하나의 대상이 여러개의 Object로 이루어졌을 경우 대상 또는 대상들의 위치, 크기를 정확하게 맞추기 위해서 이러한 기능들은 필수적이다.
Mapping (Texturing)	대상의 표현을 RGB mode로 사용자가 정의한다. 자신이 만든 이미지 또는 스캔된 이미지로 대상에 적용 시킨다.	정확한 Mapping을 위해서는 대상에 대한 이미지의 비율과 위치 관계를 알아야 한다.	대상에 대한 Mapping 소스의 크기와 비율 그리고 정확한 위치 설정을 위해서는 필수적 요소들이다.
Lighting	자연 빛은 Sun, 인공 빛은 Spotlight로 표현. 색상은 RGB mode표현. Spotlight는 빛의 중심인 Target과 Falloff로 구분	대상과 그 속성을 온전히 전해주는 요소로서 현재는 이를 이용하여 다양한 특수효과를 부여할 수 있다.	Light의 정확한 위치는 대상에 대한 온전한 표현을 가능하게 하기 때문에 이들 기능은 필수적이다.
Camera	Target이 있으며, Focal은 Camera 앵글의 범위를 정하는데 사용한다.	사용자의 주관을 표현하기 위한 수단이다.	카메라의 대상에 대한 위치 및 앵글조절은 사용자의 주관 표현과 관계
Animation	대상의 중심은 Hierarchy이며, 이는 모든 Animation 의 중심이며, 종속관계를 유지하는 중심축이다.	모든 대상은 Animation이 가능하다. 왜곡은 같은 수의 점을 가진 두 대상사이에서 이루어진다.	모든 대상은 Animation 가능하며, 이들 Animation은 좌표로 표현.
Rendering	한국등과 같이 NTSC방식을 선택한 국가에서는 초당 30프레임을 계산한다.	유럽등과같이 Pal/Secam 을 쓰는 국가들에서는 초당 25프레임을 기준	

### 3. 3D Software의 논리성

현실세계와의 비교로 3D Software는 Modeling, Texturing 또는 Mapping, Camera, Light, Animation, Rendering으로 분류

되어져 있음을 살펴보았다. 그럼 이러한 구성을 사용자는 어떻게 쉽게 이해할 수 있을까? 이를 위해서 3D Software는 어떤 방법으로 설계되어 있을까?를 우선적으로 살펴보는 것이 문제 해결에 도움이 된다. 이는 인간의 대상 인식의 논리 구조를 따르고 있다. Kant의 인식론을 따르자면, 만일 많은 종류의 집이 있는데도 불구하고 인간이 그들 중 하나의 집 - 비록 그것이 언뜻 집과 같지 않다고 할지라도 그것을 보고 집이라고 할 수 있는 이유는 우리의 인식 내에 집이라고 함은 이러한 것이라는 정의가 내재한 까닭이라고 한다. 즉 개념에 의해서인 것이다. 즉 種개념에서 시작하여 그 이하의 하부개념들이 차례로 파악되는 것이다. 이는 3D Software에서 뿐만 아니라 모든 Software에서 마찬가지이다. 즉 대개념 -> 중개념 -> 소개념으로 이루어져 있는 것이다. 여기서 워드프로세서의 예를 들고자 한다. 워드프로세서의 프로그램들은 대개의 경우 File, Edit등과 같은 대개념 하에 각각 그 자신의 하부개념을 가지고 있다. 그리고 그 중의 몇 가지의 하부개념은 또한 그 자신만의 하부개념을 갖고 있다. 그럼 왜 워드프로세서에서는 왜 'File' 이라고 불렀으며, 그 안에 왜 Open, Save와 같은 하부 개념을 포함하고 있는 것일까? 파일은 현실에서 우리가 들고 다니는 서류철을 뜻한다. 이러한 서류철에서 우리는 서류를 꺼내어 보는데 이것이 바로 워드프로세서의 경우 'Open'에 해당되어지며, 아이콘 역시 서류철 모양에 서류철이 열린 상태이다. 그리고 어떤 문서를 보고 난 후에 무엇을 적어 다시 서류철에 넣었을 때 이것이 'Save'라 하고 아이콘은 서류철이 거의 닫힌 상태에서 화살표가 서류를 넣는 쪽에 향해 있다. 이처럼 컴퓨터에 있는 하나하나의 개념들과 아이콘은 제멋대로 만들어진 것이 아니라 인간의 인식 방법과 행태를 충분히 검토하고, 그 타당성이 입증 된 후 만들어진 것이다. 이는 비록 워드프로세서 보다 훨씬 더 복잡하지만, 3D 소프트웨어에 있어서도 마찬가지로 전체적 구성은 인간의 이해에 기초하며, 아이콘 역시 마찬가지 맥락이라고 할 수 있겠다. 따라서 사용자는 3D Software의 대개념에서부터 순차적으로 그들 개념을 이해해야 하며, 그 다음 그에 해당하는 단어의 개념을 잘 이해한다면, 어느 정도 Software의 기능들을 개념적으로 이해한다고 말할 수 있겠다. 왜냐면 단어는 설명의 개념적 표현임과 동시에 이것이 사용자와 Software 프로그래머 간의 첫 번째 대화창이기 때문이다. 그 후, 구체적으로 그 기능들을 적용시키면, 대부분의 Software의 경우 대화창 하단에 그 기능에 대한 설명 내지는 지시사항 또는 사용자가 잘못 적용한 것에 대한 설명이 나타난다. 이러한 메시지는 사용자의 이해를 돕기 위한 것이기 때문에 새로운 Software를 사용하는 사람들은 간과해서는 안될 부분이다. 따라서 사용자는 처음 대하는 3D Software에서 가장 먼저 해야 할 일은 어디에서 modeling을 하고, 어디에서 animation을 하며, 어디에서 mapping을 하는가 등의 대개념의 위치들을 알아두어야 하며, 그리고 그에 해당하는 하부개념에 대한 개념적 이해가 선행되어야 하며, 그 다음 그 개념들 하나하나가 어떤 방법으로 어떻게 그리고 어느 부분에서 적용되는지에 대해서 알아야 할 것이다.

### 4. 결 론

이상에서 살펴 본 것처럼 3D Software는 Modeling, Mapping(Texturing), Animation, Lighting, Camera 그리고 Rendering으로 분류되어짐을 알았다. 이것이 근래에 와서는 Particles, Dynamics, Kinetics 및 여러 가지 특수효과등이 첨부되어 3D Software의 기능이 더욱 세분화, 복잡화되어 간다고 할 수 있겠다. 이러한 것들은 또 다른 설명이 필요한 것들이나, 전체적인 3D Software의 기본 구성은 상기에서 언급한 틀에서 크게 벗어나지 않는다. 따라서 이상과 같은 3D Software 일반에 대한 이해, 그리고 대개념에서 하부개념으로의 순차적 이해과정이 이루어지고, 이러한 이해 하에서 구체적인 기능을 적용 시켜 본다면, 3D Software들에 대한 사용자의 적응 능력은 매우 유연해질 것이다.

## 참고문헌

1. Immanuel Kant, Kritik der reinen Vernunft, Meiner, Februar, 1971
2. Françoise Holtz-Bonneau, Création infographique, Addison-Wesley, Novembre, 1994
3. Revue d'esthétique, Les techimage, Jean Michel Place, 1994
4. Michel Bret, Image de synthèse, Dunod, mai, 1988
5. Régis Debray, Vie et mort de l'image, folio essai, décembre, 1994