

3D 객체 표면에 근거한 Sketch 입력기법 연구

신은주¹, 최윤철², 임순범³
숙명여자대학교 멀티미디어학과^{1,3}, 연세대학교 컴퓨터과학과²
{kiki75¹,sblim³}@sookmyung.ac.kr, ycchoy@rainbow.yonsei.ac.kr²

A Study on Sketch Input Technique by Surface of 3D Object.

Eun- Joo Sin¹, Yoon-Chul Choy², Soon-Bum Lim³
Dept. of Multimedia Science, Sookmyung Women's University^{1,3}
Dept. of Computer Science, Yonsei University²

요약

3D 가상공간을 협업에 효과적으로 이용하기 위해서는 3D 가상공간에서 빠르고 쉽게 의사를 표현할 수 있는 기술이 필요하다. 본 연구에서는 이런 효과적인 의사표현 방법으로 Sketch 기법을 제안한다. Sketch 기법은 간단한 2D 선들의 표현을 통해 매우 빠르게 아이디어를 표현할 수 있으며, 의도에 따라 부분을 강조하거나 가감할 수 있기 때문에 좀 더 직관적으로 의도를 전달할 수 있다. 그래서 이러한 Sketch를 공간상에서 손쉽게 3D 모델의 표면이나 공간 위에 입력하고, 입력한 Sketch를 3D 가상공간과 연동하여 보여줌으로써 쉽게 3D 가상공간 안에 Sketch로 의견을 표현할 수 있게 하는 Sketch 기반 인터페이스를 연구하였다.

Sketch 기법을 3D 가상공간에 적용하기 위해서는 3D 가상공간 상에서 2D Sketch를 입력하기 위한 방법과 입력 시 발생할 수 있는 위상차의 문제를 해결해야 한다. 이 연구에서는 2D Sketch 입력을 대상 3D 객체 중심에서 Sketch할 부분의 Sketch plain을 선택 및 생성함으로써 입력하는 방법을 연구하였으며, 입력 시 발생하는 위상차는 View-point 및 View-Plain의 이동 등을 통해 해결점을 찾고자 하였다. 그리고 추후 연구를 통해 Sketch를 Annotation으로 활용, 협업에서 필요로 하는 Sketch Annotation으로 개발하고자 한다.

Keyword : Sketch on space, VR, Sketch based Interface, Collaboration, Sketch plain, Phase difference

1. 서론

3D 가상공간에 대한 기술이 발전하면서 다양한 분야로의 활용이 시도되고 있지만, 대부분의 연구가 3D의 사실적인 표현이나 현실감 있는 시뮬레이션에 그 초점이 맞추어져, 3D 가상공간을 실제적으로 활용하기에는 매우 미흡한 상황이다

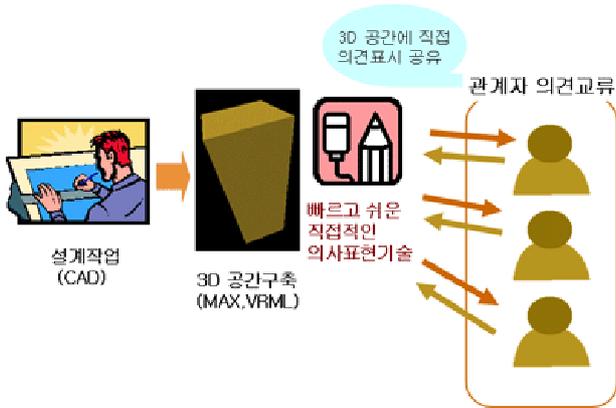
공간을 디자인한다는 점에서 건축이나 도시설계 분야에서도 3D 가상공간의 활용은 그 효율성

이 매우 높을 것으로 기대되지만, 현재 작업의 결과만을 보여주는 것에만 활용될 뿐 3D 가상공간을 효과적으로 활용하지는 못하고 있다. 그 이유는 다음과 같다. 건축이나 도시설계는 다양한 관계자들에 의해 의견을 교류, 수렴하여 수정을 반복해 감으로써 진행되는데, 현 3D 가상공간의 기술이 이러한 협업을 지원하지 못하고 있기 때문이다.

빈번히 발생할 수 있는 수정의견을 서술 또는 설계 도면으로 표시해, 3D 전문가에 의해 그때 그때 가상공간을 수정해서 보기에는 막대한 물적, 시간적 자원이 소요되기 때문이다. 따라서 건축이나 도시설계의 과정에서 협업을 지원하

본 연구는 2005년 정보통신부 IT 학술기초연구지원사업(B1220-0501-029)의 지원에 의하여 수행되었음.

기 위해서는 무엇보다 초기 설계를 기반으로 한 3D 가상공간에 직접적으로 관계자들이 빠르고 쉽게 의견을 표현할 수 있어야 한다.

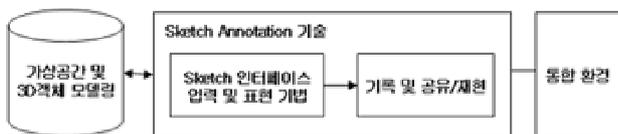


[그림 1-1] 협업을 지원하기 위한 3D 가상공간 기술

이러한 표현의 방법으로 Sketch 기법을 제안한다. Sketch 기법은 전통적으로 매우 빠르게 아이디어를 표현할 수 있으며, 의도에 따라 부분을 강조하거나 가감할 수 있기 때문에 좀더 직관적으로 의사를 표현할 수 있는 장점을 지니고 있다. 또한 Sketch의 명확하지 않은 선들은 오히려 협업이 다루는 중간과정에 사고의 폭을 높일 수 있다는 장점을 지니고 있기 때문에 협업 시스템을 위한 의사표현 기법으로 매우 적합하다.

본 연구에서는 이러한 Sketch 기법을 이용하여 협업을 지원할 수 있는 가상공간 시스템을 제안하며, 기본적인 의사표현의 도구로 3D 객체표면에 근거한 Sketch 입력기법을 연구하였다. Sketch를 공간상에서 손쉽게 3D 모델의 표면이나 공간 위에 입력하고, 입력한 Sketch를 3D 가상공간과 연동하여 보여줌으로써 쉽게 3D 가상공간 안에 Sketch로 의견을 표현할 수 있게 하는 것이다.

이 연구는 추후 입력한 Sketch를 Annotation으로 활용하여 협업을 보다 효율적으로 지원할 수 있도록 기록 및 공유/재현할 수 있는 기술을 연구, 최종적으로 문서와 가상공간을 통합한 환경에 적용하고자 한다.



[표 1-1] 연구의 최종목표 및 핵심내용

2. 관련연구

3D 가상공간을 활용하고 있는 건축이나 산업디자인 분야에서 아이디어를 직관적으로 표현하는 방법으로 Sketch 기법을 응용하는 연구는 진행되고 있으나, 이는 주로 기존의 복잡하고 어려운 3D 모델링 작업을 개선하기 위한 방법으로써의 Sketch 연구가 대부분이다.

따라서 협업과정에서 모델링 과정 외에 전반적인 과정에서의 의견교류를 위한 Sketch에 대한 연구는 아직 초기단계라 할 수 있다.

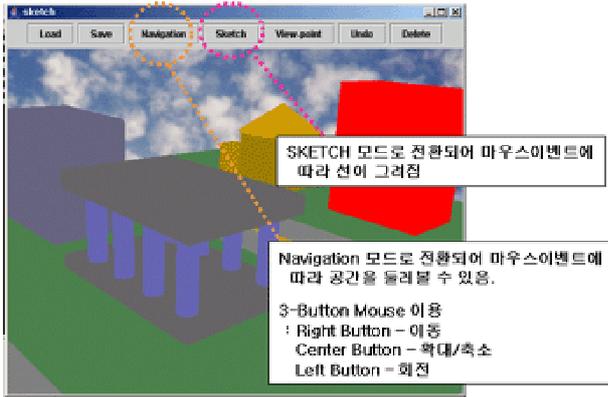
유럽의 IMAG-INRIA 연구소에서는 2D Sketch 선을 Multipass Stroke Rendering Algorithm을 통해 3D 화하여 가상공간상에 표시하는 연구를 하여, Sketch를 Annotation으로 활용할 수 있는 가능성을 제시 하였지만 Sketch 입력이 매우 제한적이었다.[1]

University of Washington에서는 Space Pen이라는 시스템을 통해 Sketch를 가상공간에서 활용하는 연구를 진행하였으나 2D Sketch를 3D 환경으로 투영시키는 과정에서의 복잡한 계산량으로 속도가 매우 느리며, Z-버퍼 알고리즘을 통해 입력할 위치를 찾는 방식으로 Sketch를 입력하고자 하는 면을 가장 앞에 위치 시키기 어려울 경우에는 불가능한 제한이 있다.[2]

State University of NewYork에서도 협업을 위한 Sketch-Based Interface를 연구하였으나 Sketch를 Annotation으로 활용하는 것으로는 발전 시키지 못하였다.[3]

3. 협업을 위한 Sketch 기반 인터페이스 시스템 제안

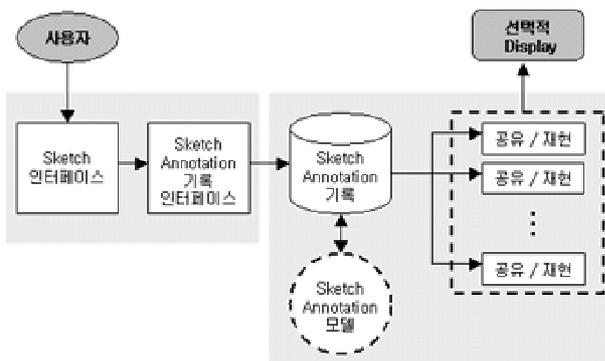
협업을 위한 시스템으로 Sketch 기반 인터페이스의 장점을 살리기 위해서는 무엇보다 간편한 조작을 통해 손쉽게 Sketch를 할 수 있어야 한다. 따라서 본 연구에서 설계, 구현 중인 시스템은 별개의 조작 메뉴 없이 마우스를 통해 네비게이션하고 Sketch할 수 있도록 두개의 모드 선택을 통해 이루어 진다.



[그림 3-1] Navigation 과 Sketch 모드에 따른 마우스 이벤트 전환

두개의 모드를 통해 사용자는 3D 가상공간을 네비게이션 하다가 손쉽게 Sketch 할 수 있다. 객체 표면에 Sketch 를 하기 위한 기법에 대해서는 다음 장들에서 언급하였다.

이렇게 Sketch 한 의견들을 xml 형태로 저장 관리 하여 Annotation 으로 활용, 협업 관계자들이 손쉽게 관리 선택적으로 확인할 수 있어야 한다. 또한 Annotation 으로 저장된 Sketch 들은 입력자 별 또는 위치별 등으로 재구성하여 확인해 볼 수 있어야 한다. 다음은 그 시스템의 개요이다.



[표 3-1] 시스템의 개요

Sketch 의 내용 중 어느 View-point 에서나 확인이 가능해야 할 즉, View-point 에 독립적인 내용의 Sketch (예를 들어 건물의 위치이동과 같은 내용)와 View-point 에 종속적인 내용의 Sketch 를 구분하는 등 효과적인 Annotation 모델을 개발하여 활용할 수 있어야 한다.

이러한 시스템의 개발을 위해 일차적으로 객체 표면 기반의 Sketch 입력기법에 대해 연구하였다.

4. 3D 공간에서의 Sketch 입력 시 문제점

3D 가상공간 상에 구축되어 있는 3D 객체 표면에 직접적으로 2D Sketch 를 입력하기 위해서는 크게 두 가지로 다음과 같은 문제점을 해결하여야 한다.

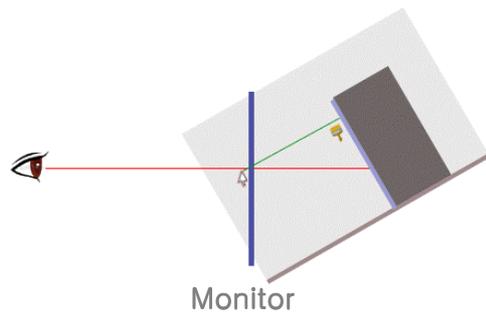
4-1. Sketch 하고자 하는 객체 면의 인식 및 Sketch 입력

3D 가상공간에 Sketch 를 입력하기 위해서는 먼저 Sketch 를 하고자 하는 면을 인식하고, 그 면에 Sketch 가 입력되도록 하여야 한다.

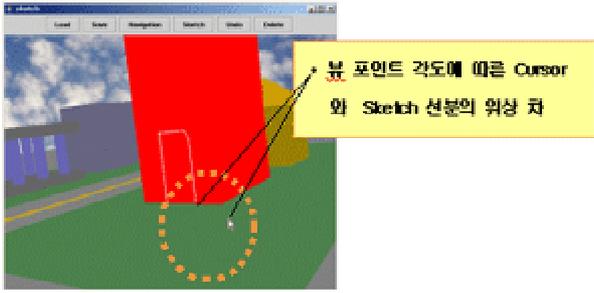
기존의 연구들에서는 주로 Z-버퍼 알고리즘을 통해 View-point 에서 가장 앞에 위치하는 면을 찾아 2D Sketch 를 투영(Projection)하는 방법을 사용하였다. 하지만 이러한 경우 Sketch 하고자 하는 면을 가장 앞에 위치 시키기 어려울 경우엔 Sketch 하는 면의 인식이 어렵기 때문에 Sketch 의 입력이 불가능하다.

4-2. 3D 가상공간에 2D Sketch 를 하게 됨으로써 발생하는 위상차

3D 가상공간에 비해 사용자가 입력하게 되는 Sketch 는 기본적으로 X,Y 축의 값만을 지니는 2D 이다. 또한 우리가 보는 Cursor 의 위치는 Monitor 로 고정된 X.Y 축이기 때문에 이러한 Cursor 의 위치 값으로 Sketch 를 생성할 경우 다음과 같은 위상차가 발생하게 된다.



[그림 4-1] Cursor 위치와 Sketch 선분의 위상차
이러한 위상차는 사용자가 Sketch 를 입력 시 Cursor 의 위치와 실제 Sketch 되는 선분과의 차이를 유발하기 때문에 자신이 Sketch 하고 있는 선의 인식을 어렵게 만들어 사실상 Sketch 가 어렵게 된다.



[그림 4-2] Cursor 위치와 Sketch 선분의 위상차의 예

기존의 연구들에서는 주로 이러한 위상차의 문제를 일종의 투영 알고리즘을 통해 해결하고자 하였으나 이 경우 상당한 계산량을 요구하여 Sketch 를 객체 면에 입력하는 속도를 상당히 떨어뜨리게 되는 단점을 지니고 있다.

5. 3D 객체 표면에 근거한 Sketch 입력 기법 제안

위 4 장에서 제기한 문제점들을 해결하고, 효율적으로 3D 객체 표면에 Sketch 를 입력하기 위한 기법으로 Sketch plain 생성을 통한 Sketch 의 입력을 제안한다. 또한 위상차를 해결하기 위해서 View-point 및 View-plane 의 이동을 제안하여 비교적 간단한 계산량으로 위상차를 해결, 속도 저하를 보완하고자 하였다.

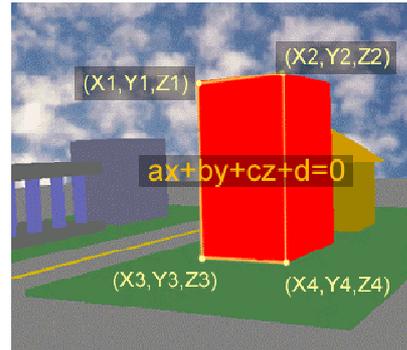
5-1. 객체표면에 근거한 Sketch plain 생성

사용자가 Sketch 를 하고자 하는 객체의 면을 선택하여 그 면을 기준으로 그 면과 동일한 Sketch plain(일종의 투명한 작업판)을 생성하여 그 면 위에 Sketch 를 하게 한다.

이러한 기법을 이용하게 되면 앞에서 언급하였듯이 Z-버퍼를 이용해 일방적으로 Sketch 를 할 경우, 제한될 수 있었던 상황에서도 Sketch 가 가능하다. 예를 들어 건물과 건물 사이의 벽처럼 면 전체를 전방에 위치 시킬 수 없어 Sketch 가 불가능할 때에도, 보이는 일부분의 객체 면 선택을 통해 그 면 전체를 인식시키고 그와 동일한 Sketch plain 을 생성 그 위에 Sketch 를 할 수 있다. 이 경우 생성된 Sketch plain 을 가리고 있는 객체의 면을 은면 처리하는 과정을 동반하

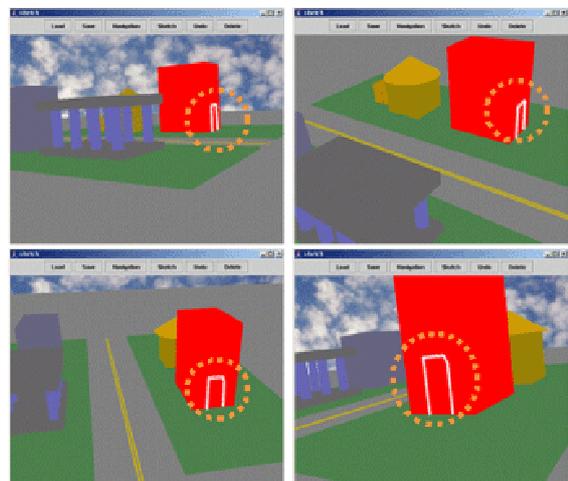
여야 한다.

Sketch 하고자 하는 객체 면의 인식은 사용자가 마우스 이벤트(Click)를 통해 면을 선택하고, 시스템은 그 객체에 해당하는 노드를 찾아 아래 그림과 같이 그 면을 이루는 꼭지점의 좌표 값을 가져오게 된다. 그리고 이렇게 얻어진 좌표 값은 외적을 통해 법선벡터와 면의 수식을 구할 수 있게 된다.



$$\begin{aligned} \text{법선벡터 } h &= ((y_2 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1), \\ & (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1), \\ & (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)) \\ &= (a, b, c) \\ \text{면의 수식은 } & a(x - x_1) + b(y - y_1) + c(z - z_1) = 0 \\ & d = -(ax_1 + by_1 + cz_1) \\ \text{즉, } & ax + by + cz + d = 0 \end{aligned}$$

[그림 5-1] 좌표 값을 통한 객체면의 인식면을 이루는 좌표 값과 면의 수식이 인식됨으로써 이를 토대로 이와 동일한 Sketch plain 을 생성할 수 있다. 그리고 이러한 Sketch plain 의 수식은 X, Y 좌표 값으로 입력되는 Sketch 를 X, Y, Z 의 좌표 값을 갖도록 변환해 준다.



[그림 5-2] 다양한 View-point 에서의 Sketch 또한 Sketch Plain 에 입력되는 Sketch 선분은

Shade3D 로 생성 함으로써 위의 그림과 같이 어느 View-point 에서도 확인이 가능하도록 하였다.

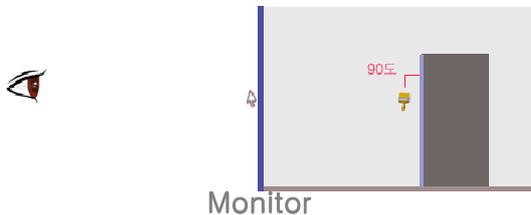
5-2. 위상 차 해결을 위한 기법

4 장에서 언급하였던 위상차 문제를 해결 하기 위한 기법으로 View-point 의 이동 과 View-plane 의 이동을 제안한다. 이 경우 매우 간단한 처리를 통해 위상차를 해결 또는 그의 오차를 줄 일 수 있기 때문에, 기존에 투영(Projection)하 는 방식의 계산량을 줄여 속도를 줄일 수 있다.

두 기법의 내용은 다음과 같다.

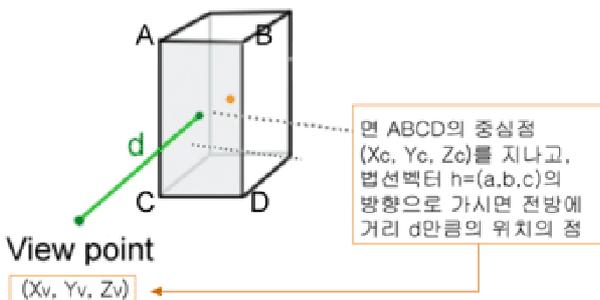
- View-point 의 이동

Sketch 하고자 하는 면을 Monitor 와 평행하게 이동시킴으로써 Curser 와 Sketch Stroke 을 한 위치로 인식하게 하여 Sketch 하도록 한다. 이 경우 현재 사용자의 위치(View-point)를 이동해야 한다는 단점이 있지만, 보다 정교한 Sketch 의 경우에 적합하며, 간단한 계산량으로 Sketch 를 입력 할 수 있다.



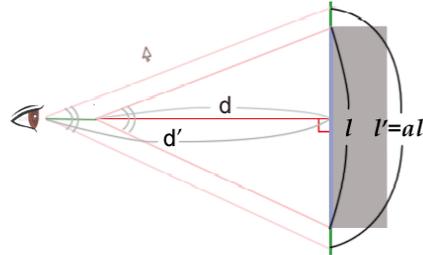
[그림 5-3] View-point 의 이동

이때 View-point 는 선택된 객체 표면 중심점 으로부터 거리 d 만큼의 직교하는 점으로 이동하 게 된다. 다음 그림에서처럼 선택한 객체면의 가 시면 전방에 View-point 를 찾아 이동하게 된다.

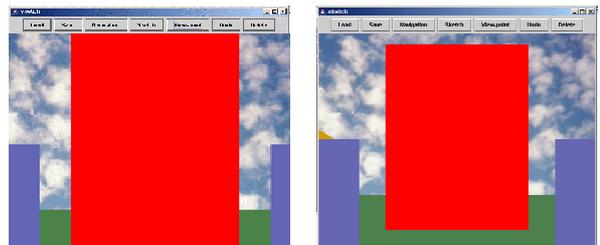


[그림 5-4] 이동할 View-point 의 위치

이때 거리 d는 면의 크기(가로, 세로 중 더 긴 쪽 길이) 에 기준하며, 면의 형태를 좀 더 잘 파악할 수 있도록 (면의 테두리까지 시야에 들어올 수 있도록) 가중치 α 값을 설정 d' 를 구한다. 예를 들어 사용자가 선택한 객체의 면을 전체 스크린 사이즈에 약 80%정도로 보고자 할 때 가중치 α 는 1.25가 되는 것이다.



l = 객체표면의 길이
 $d = l$ 을 한눈에 보기 위한 거리 d
 α = 시야 폭 조절을 위한 가중치



A. 거리 d 인 경우의 시야 B. 가중치 α 에 의해 조정된 거리 d' 인 경우의 시야

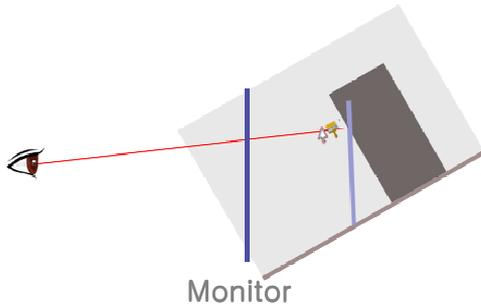
[그림 5-4] View-point 와 Sketch plain 간의 거리 d 의 설정

α 를 통해 조정한 시야이다. 사용자가 선택한 객체 면의 형태의 파악하기에 B 가 더 효과적인 것을 알 수 있다. View-point 와 Sketch plain 간의 다른 객체가 존재할 경우가 발생 할 수 있다. 즉, 건물과 건물 사이의 벽과 같이 View-point 이 동 시 시야확보가 어려울 경우인데, 이 경우는 View-point 이동과 동시에 선택된 면의 깊이 (depth)를 체크 하여, 그 사이의 객체들을 일시적 으로 가려주는 방법을 통해 해결이 가능하다.

- View-plane 의 이동

위상차 해결을 위한 기법으로 View-point 이동 과 함께 제안하는 기법은 View-plane 의 이동이다. View-point 의 이동은 간단한 전처리를 통하여 위

상차 없이 Sketch 를 입력할 수 있지만, 현재 사용자가 위치하는 View-point 를 이동해야 하는 단점이 있다. 따라서 때에 따라 사용자가 현재 위치에서 View-point 의 이동 없이 바로 간단한 Sketch 를 할 수 있도록 View-plane 의 이동을 제안한다.



[그림 5-] View-plane 의 이동

View-plane 의 이동이란 Sketch 가 입력되는 평면을 선택된 객체의 중심점 위치로 Z 값 (depth)를 주어 위의 그림과 같이 이동시키는 것이다. 이 경우 위상차를 완전히 해결할 수는 없지만 객체 면과의 Z 값의 차이를 줄여주기 때문에 간단한 계산량으로 위상차를 줄일 수 있다. 따라서 사용자가 어느 위치에서든 정교하지 않은 간단한 Sketch 는 가능하게 되는 것이다.

이 기법은 현재 Sketch 후 객체면에 Sketch 를 적은 오차로 매핑할 수 있는 방법에 대해 연구 중이다.

6. 결론 및 향후 연구

본 연구에서는 협업을 위한 Sketch 기반 인터페이스 개발을 위한 초기 연구로 객체 표면에 근거한 Sketch 기법을 제안하였다.

객체 표면에 Sketch 를 입력하기 위한 객체 면의 인식 및 위상차의 문제를 연구하였으며, 그 해결책으로 객체면에 View-point 및 View-plane 의 이동을 통해 위상차를 해결하는 기법을 제안하였다. 이 기법은 기존의 많은 계산량을 지니는 복잡한 알고리즘을 보다 간단한 방법으로 해결함으로써 제한되는 경우 없이 속도를 향상시킬 수 있다.

또한 향후 평면의 형태가 아니거나 여러 객체

로 구성된 면 위에 Sketch 를 입력하는 기법과 Sketch 를 Annotation 으로 기록, 협업 관계자들이 공유, 재현할 수 있는 기술에 대해 연구해 나가고자 한다.

7. References

- [1] Drawing for Illustration and Annotation in 3D [David Bourguignon, iMAGIS–GRAVIR/IMAG–INRIA] EUROGRAPHICS (2001)
- [2] Space Pen - Sketching annotations in a 3D Web environment [Thomas Jung , University of Washington] CHI 2002 conference
- [3] A Sketch-Based Interface for Collaborative Design [Zhe Fan, NY] EUROGRAPHICS (2004)[4] SKETCH [R. Zeleznik, Brown University] SIG-GRAPH (1996)
- [5] Teddy : A Sketching Interface for 3D Freeform Design [Takeo Igarashi, Tokyo University] 1999
- [6] A Pen-Based Freehand Sketching Interface for Progressive Construction of 3D Objects [M. Masry, D. Kang and H. Lipson, Cornell University] (2005)