# 동시 시점 변경과 시각 피드백을 제공하는 포인팅 기반 네비게이션 기법에 대한 연구

박하영, 최윤철
연세대학교 컴퓨터과학과

# A Study on Pointing-based Navigation Technique with Visual Feedback Supporting Simultaneous View Change

Ha-Young Park, Yoon-Chul Choy
Dept. of Computer Science, Yonsei University
E-mail : mint96@rainbow.yonsei.ac.kr, ycchoy@rainbow.yonsei.ac.kr

**요 약**

3D 가상환경에서의 네비게이션 기법은 공간에 대한 참여자의 몰입감과 사실감에 결정적인 영향을 미치는 매우 중요한 요소이다. 네비게이션 분야에 대한 기존의 연구는 주로 강력한 하드웨어를 기반으로 하는 몰입형 가상현실 분야 위주로 진행되어 왔다. 최근에는 인터넷을 기반으로 하는 비 몰입형 가상환경 분야에 대한 관심이 고조 되고 있으나 참여자가 인지하는 3D-2D간의 공간적 불일치 문제를 해결해야 하는 어려움을 가지고 있다. 따라서 본 논문에서는 비 몰입형 가상환경에서 참여자의 3D 공간에 대한 인지적인 문제를 해결해 줄 수 있는 네비게이션 기법으로서, 포인팅에 기반 한 3D-2D간의 매핑, 이동 방향이나 목표지점을 제시하는 시각 피드백, 이동 중 참여자 시점 동시변경 기능들을 제안한다. 제안된 기법은 3D 가상환경에서 일반 참여자의 네비게이션을 매우 효과적으로 지원할 수 있음을 확인 하였다.

## 1. 서론

3D 사이버스페이스는 3D 컴퓨터그래픽을 처리할 수 있는 컴퓨터 하드웨어의 성능이 발전하고 가상현실 저작을 위한 VRML과 같은 표준 언어의 등장 및 3D 저작을 위한 다양한 매체의 등장으로 인해 발전하고 있다. 또한 일반 사용자도 특별한 장비 없이 가상현실을 쉽게 체험할 수 있는 기반이 제공되고 네트워크의 대역폭의 증가와 인터넷의 대중화로 인해 3D 사이버스페이스에 대한 관심이 높아지고 있다.

3D 사이버스페이스는 2D 이미지에 비해 임장감과 몰입감을 제공하고 현실감을 더욱 증가시킬 수 있다는 장점이 있다. 또한 인터넷이 갖는 범용적이고 대중적인 특징과 더불어 많은 잠재적인 가능성을 갖고 있어 여러 분야에 활용되고 있다.

이러한 사이버스페이스에 사용자들이 참여할 수 있는 방법으로 여러 가지 사용자 인터페이스가 제시되고 있고 가장 대표적인 사용자 인터페이스는 이동이다. Travel은 3차원 환경하에서 사용자의 시점의 변화(Viewpoint motion)을 통제하는 것이고 길 찾기(Wayfinding)은 경로를 판단하는 인식과정인데 travel과 길 찾기를 통합한 것이 이동이다. 현재까지 이동을 위한 연구가 여러 가지 있으나 부적절한 사용자 인터페이스의 사용과 다양한 상호작용적 요소의 부재로 인하여 성공적인 활용사례가 저조한 편이다. 따라서 실세계서와 같은 사실감을 증가 시킬 수 있는 다양한 상호작용 기법에 대한 연구가 필요하다.

## 2. 관련연구

### 2.1 Input Technique for Navigation

3D 공간에서 이동을 하기 위해서 사용되는 입력 장치는 여러가지가 있지만 그 중에서 사용자들이 가장 보편적으로 사용하는 입력장치는 마우스와 키보드이다. 그러므로 마우스와 키보드를 사용해서 어떤 다른 방법으로 이동을 가능하게 하는지에 대해서 알아본다.

마우스만을 사용하면 마우스 드래깅(dragging)과 클리킹(clicking) 방법을 사용한다. 마우스를 dragging하면 drag된 방향과 길이에 따라 이동하는 방향과 속도가 조절된다. 그리고 clicking을 하면 사용자가 선택한 곳으로 시점이나 아바타가 이동하는 것이다.

키보드만을 사용하여 네비게이션하는 방법도 있는데 이때 대부분 위, 아래, 왼쪽과 오른쪽 화살표가 있는 키를 사용하여 이동한다.

대부분의 게임에서는 마우스와 키보드를 동시에 사용하는 방법을 선택한다. 마우스로는 시점을 이동하는데 사용되며 키보드로는 아바타가 이동하는 입력장치로 사용된다. 이때의 장점은 시점과 아바타의 위치를 동시에 조작할 수 있다는 것이다.

### 2.2 Visual Feedback for Navigation

가상환경에서 사용자의 편의를 위하여 커서를 조작하는 방법에 대한 연구가 여러 가지 있다.

첫째로 [2]는 3D 커서를 사용하는 방법을 제안하고 있다. 3D 커서는 오브젝트가 어느 방향으로 조작되고 있는지 알게 한다. 이 논문에서 제시한 다른 커서 조작은 어떤 오브젝트로 인해 커서가 보이지 않게 되면 커서를 가리고 있는 오브젝트를 투명하게 하는 방법을 제시하고 있다.

[3]은 커서를 십자(crosshair) 모양으로 하는 방법을 제시한다. 이것의 목적은 [6]에서와 같이 오브젝트가 어느 방향으로 조작되고 있는지 더 쉽게 알 수 있도록 하는 것이고 항상 x, y, z축이 보이므로 어느 방향으로 오브젝트가 이동하고 있는지 사용자가 눈으로 더 쉽게 확인할 수 있다.

위에서 제시한 방법들은 오브젝트의 조작 방향을 더 쉽게 눈으로 확인할 수 있다는 장점이 있지만 crosshair나 3D 커서는 너무 많은 공간을 차지하기 때문에 불편한 점이 있다.

## 3. 네비게이션 기법

본 논문에서 제시하는 네비게이션 기법은 입력, 처리, 출력 과정을 모두 다루고 있다.

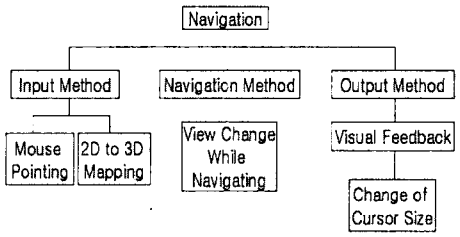


그림 1. 제안하고 있는 기법들의 도표

### 3.1 Pointing-based Navigation

본 연구에서는 마우스를 click함으로서 이동하는 네비게이션 방법을 제시한다. 마우스를 drag하여 이동하는 방법은 속도를 조절하는데 어려움이 있고 계속해서 마우스 버튼을 눌러야 하므로 손에 피곤을 줄 수 있기 때문에 마우스를 pointing 하는 방법을 사용한다.

● 2D to 3D Mapping

우선 마우스로 어느 지점을 선택했을 때의 좌표는 스크린 상의 좌표이기 때문에2D이지만 실제로 아바타가 이동해야 하는 좌표는 3D 이므로 2D 좌표를 3D로 매핑하는 과정이 필요하다.

이렇게 하기 위한 식을 정의하기 전에 몇 개의 가정을 한다. 사용자가 선택한 좌표는 (x, y)이고 아바타가 이동해야 하는 좌표는 (x', y', z')라고 한다. 그리고 지형이 평면이라고 가정하여 3D 좌표에서의 y는 0으로 고정한다.

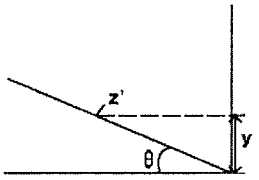


그림 2. 3D 워드와 스크린의 옆 모습

그리고 z는 사용자가 선택한 좌표인 (x, y)에서 y의 영향을 받아 다음과 같은 식이 성립된다.

$$z' = \frac{y}{\cos\theta}$$

(1)

(1)에서 θ는 가상환경이 스크린에 투영된 각도이다.

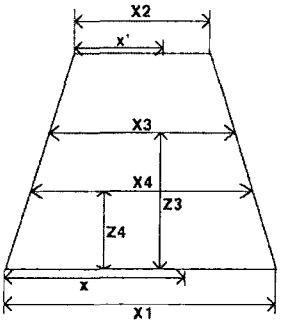


그림 3. 3D 워드와 스크린의 앞 모습

x'는 위에서 구한 z' 값과 사용자가 마우스로 선택한 점 (x, y)에서 x 값의 영향을 받아 다음과 같은 식을 사용하여 구할 수 있다.

$$x' = \frac{x' \cdot \alpha \cdot z'}{X1}$$

(2)

여기서 $\alpha$는 투영에 의한 상수이고 X1은 스크린 가장 앞쪽 부분의 길이이다.

### 3.2 Simultaneous View Change

이미 존재하는 많은 게임들은 시점을 변경하기 위해서 마우스를 사용하고 위치 변경을 위하여 키보드를 사용하는 방법을 선택하고 있다. 하지만 이 같은 방법은 시점을 이동하면서 아바타를 움직이는데에는 한계가 있다. 또한 마우스만을 사용하여 이동하면서 동시에 시점을 변경하는 기법에 대한 연구가 미비하다. 따라서 본 논문에서는 네비게이션하면서 동시에 시점 변경이 가능한 방법을 제시한다.

앞에서 이미 제시한데로 이동할 때는 이동하고 싶은 위치를 마우스로 클릭킹을 하면서 선택한다. 아바타가 그 위치로 이동하고 있는 동안 아바타가 이동하고 있는 경로 주위에 무엇이 있는지 확인 할 수 있도록 하기 위하여 마우스가 움직이면 시점이 변경되는 방법을 선택하였다.

아바타가 이동하고 있을 때 마우스가 왼쪽으로 이동하였으면 시점 또한 왼쪽으로 이동하고 오른쪽으로 이동하면 시점도 오른쪽으로 이동한다. 키보드와 마우스를 사용하여 시점과 이동 위치를 변경하는 경우에는 시점이 변경되면 이동 경로 또한 변경되지만 본 논문에서 제시한 방법을 사용하면 경로는 변경되지 않고 다만 시점만이 변경된다.

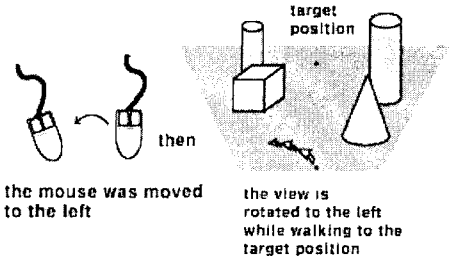


그림 4. 시점 변경을 위한 기법

### 3.3 Visual Feedback for Navigation

사용자가 가상환경에서 마우스를 사용할 때 느끼는 가장 큰 불편은 마우스가 어느 위치로 이동하는지 파악하는 것이 어렵다는 점이다. 가상공간은 3D 이지만 스크린이 2D이어서 커서가 네비게이션 가능한 방향은 x와 y축이기 때문에 문제가 생긴다. 이러한 문제점을 해결하기 위하여 커서의 크기를 조작하는 방법을 제안한다.

커서가 이동할 때 아바타의 현 시점에서부터 멀어지면 커서의 크기가 작아지고 가까워지면 크기가 커지는 효과를 주어 커서가 어느 축으로 이동하고 잇는지를 사용자가 눈으로 확인할 수 있게 한다.

네비게이션 기법에서 이미 말한 바와 같이 평면으로 이동한다고 가정하여 마우스는 y 축으로 이동하지 않는다고 가정한다. 그리고 커서 크기를 변화하기 위해 사용되는 식은 2D 좌표를 3D 좌표로 매핑하는데 사용되는 식과 동일하다.

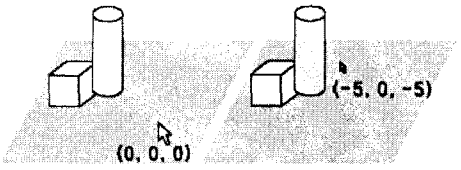


그림 5. 커서 크기의 변화

## 4. 구현 및 고찰

본 시스템은 Windows 2000 환경에서 C++, MFC, OpenGL를 이용하여 구현하였다. 본 시스템은 향후 네비게이션하는 속도가 빠를 필요가 없고 주위에 무엇이 있는지 자세히 알아야 하는 사이버 쇼핑몰에 사용될 수 있을 것으로 예상된다.

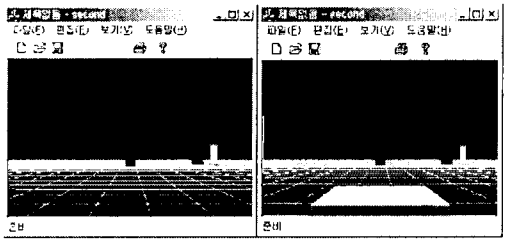

그림 6. 제안한 네비게이션 기법을 적용한 시스템

본 논문에서 제시한 기법은 공간에 있는 사물에 의해 시야가 가려질 때 이동하는 경우에 효율적이다. 기존의 방법을 사용할 때는 목적지까지 도착하거나 현재 진행 중인 이동을 멈추지 않고서는 시점을 변경하여 다른 곳을 선택할 수밖에 없었다.

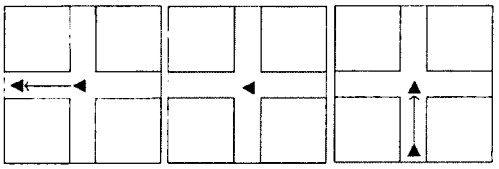

그림 7. 기존의 기법을 사용한 네비게이션

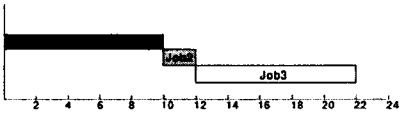


그림 8. 기존의 방법의 네비게이션 시간

하지만 본 논문에서 제시한 방법을 사용하면 이동하면서 주위에 있는 환경을 볼 수 있어서 일차적으로 사용자가 선택한 위치로 이동하는 중에 다른 목적지를 선택하여 네비게이션 방향을 변경할 수 있기 때문에 시간도 조금 걸리고 더 효율적이라고 볼 수 있다.

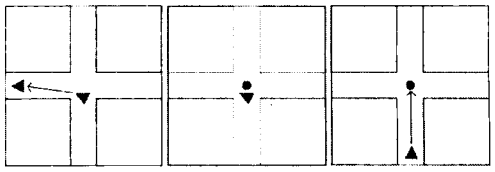

그림 9. 제시한 기법을 사용한 네비게이션

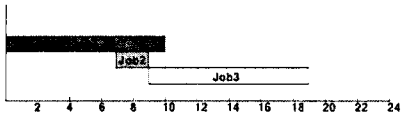


그림 10. 제시한 방법의 네비게이션 시간

## 5. 결론

본 논문에서는 마우스를 이용하여 3D 공간에서 네비게이션하는 방법에 대해서 연구하였다. 입력으로는 마우스의 pointing을 사용하여 이에 선택된 2D 좌표를 3D 좌표로 매핑하는 방법에 대해서 알아봤다. 그리고 이동하면서 동시에 시점을 변경하는 방법과 커서의 크기를 조절하여 사용자에게 visual feedback을 제공하는 방법을 제안하였다.

향후 연구계획으로는 현재 y축에 대한 조작이 가능하지 않으므로 평면이 아닌 곳에서 네비게이션할 수 있는 방법과 위, 아래를 볼 수 있도록 하는 방법에 대해 연구한다. 또한 좀 더 효율적인 네비게이션 방법이 될 수 있도록 이미 제시한 네비게이션 방법에 충돌 방지 기법을 제공하는 연구 또한 필요하다.

## [참고문헌]

[1]Chen, Mountford, and Sellen, "A Study in Interactive 3D Rotation Using 2D Control Devices," Computer Graphics, 22(4):121-128, 1988

[2]Stein, T., and Coquillart S., "The Metric Cursor", in Computer Graphics and Applications, 2000. Proceedings. The Eigth Pacific Conference on, 2000

[3]Dennerlein, J.T., Martin, D.B., and Hasser C., "Force-Feeback Improves Performance for Steering and Combined Steering-Targeting Tasks", Proceedings of the CHI 2000 conference on Human factors in computing systems, 2000, Pages 423-429

[4]Dongo Xiao and Roger Hubbold, "Navigation Guided by Artificial Force Fields", in Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems, Vol. 1 pp. 179-186, April

1998

[5]Parris K. Egbert and Scott H. Winkler, "Collision-Free Object Movement Using Vector Fields", IEEE Computer Graphics and Applications, vol. 16, no. 4, pp. 18-24, July, 1996