

Leap Motion을 이용한 새로운 제어 인터페이스 기술 개발

김현택, 남재현, 정우환, 조국진, 김병규
선문대학교 컴퓨터공학과
e-mail: bg.kim@mpcl.sunmoon.ac.kr

Development of New Control Interface Technology Based on Leap Motion

Hyun-Taek Kim, Jae-Hyun Nam, Woo-Hwan Jung, Kuk-Jin Cho,
Byung-Gyu Kim
Dept of Computer Science and Engineering, SunMoon University

요 약

최근 정보기술의 발전에 가속화되면서 기존의 키보드나 마우스 같은 전통적인 제어장치들과는 달리 새로운 하드웨어 인터페이스들이 등장하고, 그에 따른 새로운 시각에서의 제어 방법들 등장하고 있다. 본 논문에서는 Leap Motion이라는 새로운 하드웨어 인터페이스를 사용할 때 제공되는 단순한 위치 정보만 쓰는 것이 아니라 어떠한 위치에 있는지, 특정 동작을 취했을 때 모션 등과 같은 정보들도 함께 사용하기 위해 손과 손가락을 인식할 수 있도록 모션 인식방법을 제안하고, 제안된 기술을 기반으로 3D Avatar를 제어 및 편집 후 애니메이션을 만들어 공유할 수 있는 모션제어 저작SW를 개발하였다. 본 기술은 의료, 게임, 교육 등 다양한 분야에서 새로운 제어 기술로서 활용될 수 있으며, 기존의 공간 터치 기술과는 다른 장점을 제공할 것으로 예상된다.

1. 서론

Graphic User Interface (GUI)가 개발되고 마우스가 광범위하게 사용되기 시작한 후 10여 년 동안 컴퓨팅 상호작용의 패러다임은 대체로 변함이 없었으며, 키보드와 스크린 상의 포인터만 있으면 어떤 식으로든 제어가 가능하였다. 기존 컨트롤러부터 벗어나, 새로운 인터페이스로 향하게 하는 변화의 조짐이 만들어 지면서 새로운 하드웨어 인터페이스 설계자들이 잇따라 등장함에 따라, 기대감을 높이고 있다 [1]. 이러한 흐름 속에서 다양한 제어 기술에 대한 시도들이 보고되고 있다 [2]-[4].

마이크로소프트의 Kinect는 RGB인식 카메라와, 적외선 카메라, 적외선 프로젝터를 사용하여 사용자의 20개의 관절 X, Y, Z축의 정보를 알려주는 제품이다 [2]. MYO는 착용자의 근육 안에 흐르는 전기 신호를 인식하여 손을 흔들거나 손가락을 움직이고, 팔을 돌리는 등의 움직임 정보를 알려주는 제품이다 [3]. 또한, Kinect처럼 깊이를 구분하는 적외선 카메라를 사용하여 사용자의 각각의 손바닥과 손가락의 X, Y, Z축 정보를 알려주는 기술을 제공한다 [4].

이러한 하드웨어의 발전과 함께 다양한 제어기술들이 제안되고 있다. Jeon et al.은 Kinect의 깊이 센서를 이용한 핸드마우스를 개발하였다 [5]. Kinect를 통해 손을 인식하여 이동한 거리를 계산하여 마우스의 위치를 이동 시키고, 손가락의 모양으로 클릭 이벤트를 구현하였

다. Lee et al.은 Kinect를 활용한 요가 학습 콘텐츠를 개발하였다 [6]. Kinect의 모션 인식은 손 제스처 인터페이스 구축과, 3D 모델 애니메이션을 가능하게 하여 요가 학습의 흥미를 유발시키고, 튜터 영상과, 구분 동작에 의한 학습, 그리고 기존 동작과 사용자 동작의 1:1 매칭 시스템으로 사용자가 정확한 동작을 학습 할 수 있도록 돕는다.

Kinect로 손과 손가락의 움직임을 인식하여 3D 가상공간에서 이동하고 물체를 선택하는 시스템이 개발되었고 [7], 손의 제스처를 인식하여 로봇을 제어하는 응용 프로그램도 제안되고 있다 [8].

이렇듯, 모션 정보에 대한 관심이 많아지면서 많은 제품들이 출시를 앞두고 있고, 모션 컨트롤러의 공간 지각적인 기능을 이용한 제어, 콘텐츠 제공 등 다양한 분야에서 기대 효과가 클 것으로 보고 있다.

본 논문에서는 Leap Motion이 사용자의 손과 손가락의 구분 없이 X, Y, Z축에 대한 위치 정보만 제공되기 때문에 양 손을 구분하고 각 각의 손가락을 구분해야 하는 경우 오른손과 왼손을 인식하는 방법과 각 각의 손가락을 인식하는 알고리즘을 제안하고, 3D Avatar를 제어, 편집하여 공유할 수 있는 콘텐츠 저작 SW를 개발하였다.

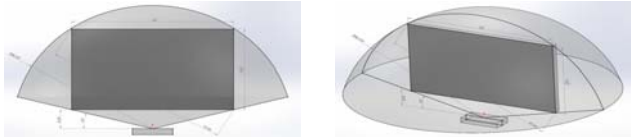
2. Leap Motion System 소개

Leap Motion System의 모습은 그림 1과 같다. 깊이를 구분하기 위한 적외선 카메라를 이용하여 3차원 스캔을

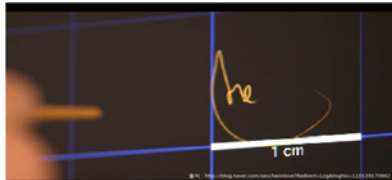
가능하며, 사용자의 손바닥과 손가락의 정보를 Leap Motion에서 제공하는 개발지원SW를 이용하여 가공이 가능하다. 그림 2와 같이 디바이스 +15°부터 구형으로 최대 초당 290 프레임 인식, 그림 3는 0.5m 반경으로 된 반구 영역에서 0.01mm x 0.01mm를 인식할 수 있는 정밀성을 제공해 준다.



(그림 1) Leap Motion



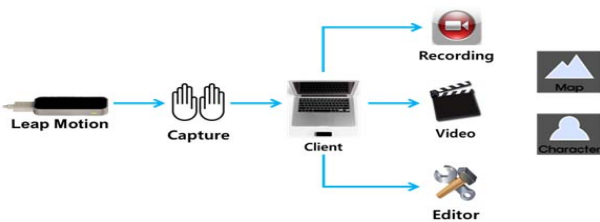
(그림 2) Leap Motion의 인식 범위



(그림 3) Leap Motion의 정확도

3. 개발된 모션 제어시스템 흐름도

그림 4는 개발된 모션 제어시스템의 흐름도를 나타낸 것이며, Leap Motion으로부터 사용자의 손과 손가락에 X, Y, Z축 위치 정보를 얻어와 Client에서 오른손, 왼손으로 인식 하는 과정(5.1절), 각 각의 손가락을 인식하는 과정(5.2절)을 통해 3D Avatar를 제어 하여 녹화, 편집 하고 동영상으로 내보낼 수 있다.



(그림 4) 개발된 시스템 흐름도

4. 개발된 모션 제어시스템 구성

개발된 시스템 구성은 그림 5와 같으며, .Net Framework 4.0과 XNA Game Studio 4.0, Leap Motion SDK 0.8.0을 기반으로 사용자의 Motion을 통해 3D Avatar를 제어, 편집하고 재생하는 Motion Control Engine과 추가적으로 Audio를 제어, 편집하고 재생할 수 있는 Audio Control Engine으로 구성되었으며, 개발 환경은 표 1과 같다.



(그림 5) 개발된 시스템 구성도

<표 1> 개발 환경

OS	Windows 7
Tools	Visual Studio .net 2010, Blender
Dependency	Leap Motion SDK 0.8.0

5. 제안된 Leap Motion기반의 인식 기법

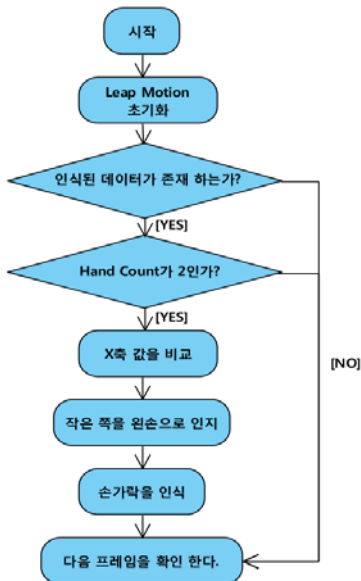
그림 6과 같이 장치의 가운데를 기준으로 X, Y, Z축에 대한 위치 정보를 획득 후 오른손 또는 왼손 구분 정보와 각각의 손가락에 대한 구분 없이 위치 정보만 있기 때문에 양 손을 구분하고 각각의 손가락을 구분해야 하는 경우, 어떻게 처리해야하는지 정해야 할 필요성이 있다. 본 연구는 SDK를 통해 가공되지 않은 Raw Data를 양 손을 인식하는 과정(5.1절)과 각각의 손가락을 인식하는 과정(5.2절)을 통해 자체 정의 클래스로 구분 하였다.



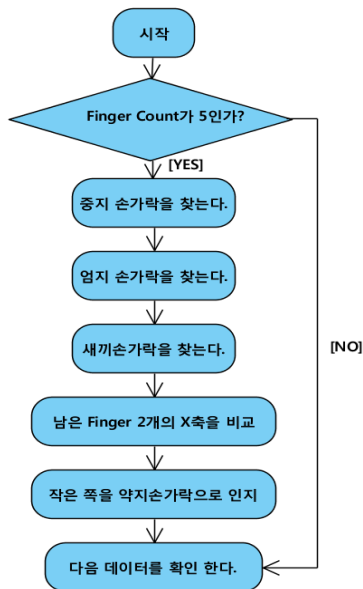
(그림 6) Leap Motion의 위치 정보 인식 기준

5.1 양 손의 인식

그림 7은 Leap Motion으로부터 얻은 데이터를 이용하여 양손을 인식하는 알고리즘의 흐름도이다. Leap Motion에서 얻은 데이터에서 HandList 클래스에 인식된 손의 개수가 2개 인지 확인하고, 각각의 Hand 클래스에 보관되어 있는 위치 정보들 중에 X축 값을 비교하여 작은 쪽의 Hand클래스를 왼쪽, 다른 Hand클래스를 오른손으로 인지한다.



(그림 7) 양 손 인식 흐름도



(그림 8) 왼손 일 때 손가락 인식 흐름도

5.2 손가락 인식

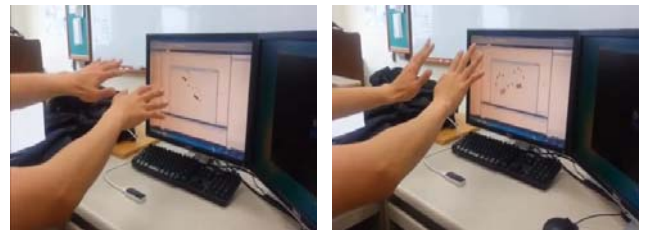
그림 8은 Leap Motion으로부터 얻은 데이터를 양손으로 인식한 후, 하나의 Hand 클래스에서 각각의 손가락을 인식하는 과정이다. 먼저, 양 손 인식 과정을 거친 하나의 Hand 클래스를 왼손이라고 가정하면, Hand 클래스에 들어 있는 손가락의 수가 5개인지 확인한다. 손가락의 위치 정보들이 저장되어 있는 FingerList에서 5개의 Finger 클래스들 중 Z축의 값이 가장 작은 Finger 클래스를 중지손가락으로 인지한다. 다음으로, 5개의 Finger 클래스들 중 X축의 값이 가장 큰 Finger 클래스를 엄지손가락으로 인지하고, 5개의 Finger 클래스들 중 X축의 값이 가장 작은 Finger 클래스를 새끼손가락으로 인지한다.

남은 2개의 Finger 클래스의 X축을 비교하여 작은 값을 갖는 Finger 클래스를 약지 손가락으로 인지하고, X축이 큰 쪽의 Finger 클래스를 검지손가락으로 인지한다. 오른손 같은 경우, Z축 값이 가장 작은 Finger 클래스를 중지손가락, X축 값이 가장 큰 Finger 클래스를 새끼손가락, X축이 가장 작은 Finger 클래스를 엄지손가락, 남은 2개의 Finger 클래스들 중 X축을 비교해 작은 값을 갖는 Finger 클래스를 검지손가락으로 구별하고 큰 값을 갖는 Finger 클래스를 약지 손가락으로 인지하도록 하였다.

6. 실험 결과 및 분석

6.1 양 손 인식 실험 결과

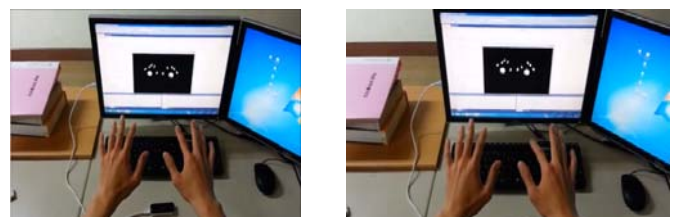
그림 9는 Leap Motion을 사용해 얻은 데이터로 양 손을 인식하는 실험의 사진이다. 사용자의 양 손의 위치 정보를 얻어와 인식하여 3D Object를 제어하는 모습이다. Leap Motion은 X, Y, Z축의 위치 정보뿐만 아니라, 기울기의 값도 얻어 올 수 있기 때문에 그림 9(a)와 같이 오른쪽으로 기울었을 때 3D Object도 같이 오른쪽으로 기울어진 것을 볼 수 있다. 그림 9(b)는 양 손을 뒤쪽으로 기울였을 때 생성된 3D 객체도 같이 뒤 쪽으로 기울어진 모습을 볼 수 있다.



(a) 오른손 기울임 (b) 양손 뒤쪽으로 기울임
(그림 9) 양 손 인식 실험 중인 사진

6.2 손가락 인식 실험 결과

그림 10은 Leap Motion을 사용해 얻은 데이터로 양 손을 구별하고, 각 각의 손가락을 구별하는 실험의 사진이다. 사용자의 양 손을 인식해 구별한 다음 각각의 손가락을 구별하여 3D 객체를 제어하는 모습이다.



(a) (b)
(그림 10) 손가락 인식 실험 중인 사진

6.3 3D Avatar 제어, 편집 실험 결과

그림 11은 양 손 구별과 각 각의 손가락을 구별하는 과정을 거쳐 3D Avatar를 제어 및 편집하는 개발된 저작 소프트웨어 실행 화면을 보여준다. 3D Avatar를 제어하기 위해 그림 12와 같이 각 손가락에 따라 제어하는 부분을 지정하고 손가락의 움직임으로 제어를 할 수 있다.

엄지손가락을 기준으로 인식되었을 때 3D Avatar의 상체 조종하고 인식되지 않고 손가락을 움직이면 하체를 조종할 수 있다. 또한, 양 손의 특정 모션과 기율기에 따라 3D Avatar를 회전, 걷기, 앉기 등과 같이 특정 행동을 제어할 수 있다. 제어를 한 손과 손가락의 정보가 1초당 30 프레임으로 XML파일에 저장 하는 녹화 기능으로 시간별 움직임에 대한 정보를 삭제, 부분추가 등과 같은 기본적인 편집이 가능하다.



(그림 11) 3D Avatar 제어 및 편집 실행 중인 화면

7. 결론

본 논문에서는 Leap Motion을 사용하여 얻은 데이터에 를 통해 양 손을 구분하고 각 각의 손가락을 구분해야 하



(그림 12) 모션 별 3D Avatar 제어

는 경우 오른손과 왼손을 구별하는 방법과 각 각의 손가락을 구별하는 알고리즘을 제안하였다. 제안된 알고리즘으로 단순히 손과 손가락의 정보만 쓰는 것이 아니라, 손과 손가락이 어떠한 위치나, 특정한 동작을 취했을 때 등과 같은 정보들을 이용할 수 있기 때문에 많은 것을 표현할 수 있다.

모션 획득 (Motion Capture) 기술을 이용하면 기존에 마우스와 키보드를 이용하여 컴퓨터를 사용하는 방식이 아닌 직접 손을 이동하고, 무언가를 잡고 그것을 내려놓고, 현실에서 직접 행동하는 것을 컴퓨터와 상호 작용할 수 있다. 앞으로 의료, 게임, 교육 등 다양한 분야에서 놀라운 변화를 창조하는 새로운 제어 인터페이스 기술로 진화할 것이라고 예상된다.

참고문헌

- [1] 정보통신산업진흥원 주간기술동향 1587호
- [2] Kinect for Windows Video Motion Capture for Virtual Characters
<http://www.microsoft.com/en-us/kinectforwindows/>
- [3] MYO [https://getmyo.com/Video Motion Capture for Virtual Characters](https://getmyo.com/Video%20Motion%20Capture%20for%20Virtual%20Characters)
- [4] LeapMotion <https://www.leapmotion.com/>
- [5] 전인배, 남부희, “Kinect의 깊이 센서를 이용한 핸드 마우스 구현” 대한전기학회 학술대회 논문집, Vol.2012, No.7, pp.1674-1675, 2012
- [6] 이주원, 오경수, “키넥트를 활용한 요가 학습 콘텐츠” HCI2012 학술대회 논문집, Vol.2012, No.1, pp.221-223, 2012
- [7] Mohd Fairuz Shiratuddin, Kok Wai Won, “Non-Contact Multi-Hand Gestures Interaction Techniques for Architectural Design in a Virtual Environment”, the International Conference on IT & Multimedia at UNITEN (ICIMU 2011), Malaysia, Nov 2011.
- [8] Asanterabi Malima, “A Fast Algorithm for Vision-based Hand Gesture Recognition for Robot Control”, Signal Processing and Communications Applications, pp. 1-4, 2006.