

촉각 피드백 기반의 비접촉식 멀티-포인팅 프레임워크

김현곤*, 박준석*

*한국전자통신연구원 스마트 인터페이스 연구팀

e-mail:freeschool@gmail.com, parkjs@etri.re.kr

A Framework For Non-Touch Multi-Pointing Supporting Haptic Feedback

Hyungon Kim*, Junseok Park*

*Smart Interface Research Team,

Electronics and Telecommunications Research Institute

요 약

최근 멀티-터치 디바이스에 대한 관심이 증가함에 따라서 멀티-터치 디바이스를 처리하는 시스템 프레임 워크도 같이 연구, 개발되고 있다. 본 논문에서는 기존의 멀티-터치 시스템 프레임워크와는 다른 촉각 피드백을 지원하는 비접촉식 멀티-포인팅 프레임워크에 대해서 소개할 것이다. 이 멀티 포인팅 프레임워크의 목적은 다수의 사용자가 다수의 디바이스를 이용하여, 시스템을 제어하고, 사용자의 입력에 따라서, 시스템으로부터 촉각 피드백을 받을 수 있는 소프트웨어적인 기본 환경을 제공하는 것이다. 이러한 멀티 포인팅 프레임워크를 구성하고 있는 각 파트에 대해서 살펴보고, 실제로 적용된 사례에 대해서 살펴보도록 한다.

1. 서론

최근 25년간 많은 멀티-터치 기술들이 연구되었으며 [1], 최근 들어서 멀티 터치에 관련된 기술들이 각광받기 시작했으며, 애플사의 아이폰[2]이 대중들에게 보급되기 시작하면서부터 멀티-터치에 대한 관심은 폭발적으로 증가하였다.

첫 멀티-터치 시스템은 1982년 토론토 대학의 Nimish Mehta에 의해 개발되었으며, 컴퓨터 시스템과 상호작용하기 위해 디자인되었다[3]. 그 이후, 멀티-터치 시스템은 많은 연구와 시행착오를 거듭하면서, 발전해 나아갔으며, 2007년에는 마이크로소프트사에서 Microsoft Surface라고 하는 첫 번째 상용 멀티 터치 테이블을 제작, 발표하였다 [4].

이처럼 멀티-터치 디바이스에 대한 관심이 증가함에 따라서, 멀티-터치 디바이스를 처리하는 시스템 프레임워크도 같이 연구, 개발되었다.[6-10] 대부분의 멀티-터치 기반의 시스템 프레임워크는 Table-Top User Interfaces Objects (TUIO) 프로토콜을 지원하며, 터치 방법과 터치한 상태에서의 제스처를 인식하여, 사용자가 화면상의 다채로운 인터페이스를 즐길 수 있도록 제작되었다.

많은 멀티-터치 기반의 시스템 프레임워크가 나왔음에도 불구하고, 비접촉식(Non-touch) 시스템과 사용자 행동에 반응하는 촉각 피드백을 제공하는 시스템 프레임워크는 개발실정이 미비한 것이 현재 실정이다. 멀티-디바이스 기반의 소프트웨어 제작의 선두에 있는 마이크로소프트

사의 Multi Point Mouse SDK[11]와 Multi Point TUIO[12]를 제외하고는, 멀티 포인팅 시스템 프레임워크는 거의 없다고 볼 수 있다. 뿐만 아니라, 사용자 입력에 따른 피드백을 제공하는 프레임워크는 국내외에도 존재하고 있지 않는 실정이다.

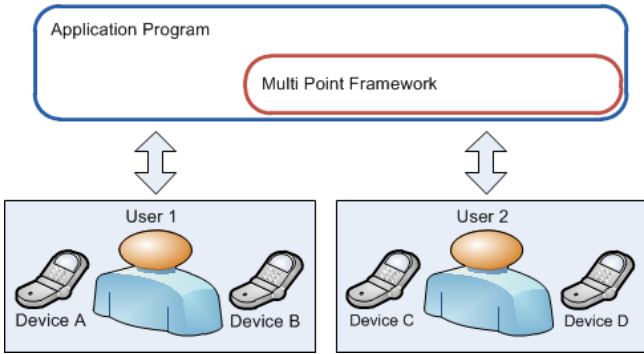
이러한 현재 실정에 비추어, 촉각 피드백을 지원하는 멀티-포인팅 프레임워크를 개발하게 되었다. 촉각 피드백 기반의 멀티-포인팅 프레임워크의 목적은 다수의 사용자가 다수의 디바이스를 이용하여, 시스템을 제어하고, 사용자의 입력에 따라서, 시스템으로부터 촉각 피드백을 받을 수 있는 소프트웨어적인 기본 환경을 제공하는 것이다.

2장에서는 멀티포인팅 프레임워크의 시스템 구조에 대해서 설명을 하고, 3장에서는 멀티포인팅 프레임워크를 사용하여 제작된 실례들을 살펴볼 것이다. 4장에는 멀티포인팅 프레임워크의 결론과 향후 방향에 대해서 논할 것이다.

2. 시스템 구조

촉각 피드백을 지원하는 멀티 포인팅 프레임워크의 사용자 및 디바이스와 시스템과의 관계도는 (그림 1)과 같다. 각 사용자는 다수의 디바이스를 가질 수 있으며, 각 디바이스는 멀티포인팅 프레임워크를 포함하는 응용 프로그램과 통신하면서 시스템과 상호작용 할 수 있다. 이 때, 각 사용자는 네트워크를 통해서 접속될 수 있으며, 한 개의 시스템에 여러 명이 동시에 접속할 수 있다. 이러한 멀티포인팅 프레임워크를 기반으로 하는 응용 프로그램은

크게 프레임워크 파트와 응용 프로그램 파트로 구분 지을 수 있다.



(그림 1) 비접촉식 멀티 포인트 프레임워크와 외부 및 내부 인터페이스와의 관계도

프레임워크 파트는 응용 프로그램에서 사용하게 될 프레임워크를 일컫는 파트로써, 간략하게 다음과 같은 순서로 처리된다. 사용자의 입력을 전달받아, 전달받은 이벤트를 분석하게 된다. 분석된 이벤트는 응용 프로그램 파트와 프레임워크 파트의 이벤트 처리부로 각각 보내진다. 응용 프로그램 파트로 전송된 이벤트는 응용 프로그램의 내부적인 처리를 거쳐 다시 프레임워크 파트로 전송되게 된다. 프레임워크로 직접적으로 전송된 이벤트는 프레임워크에 처리하도록 미리 정의된 이벤트인 경우, 해당 이벤트를 정의된 처리 순서에 따라서 각 파트로 전송하게 된다. 그 외의 이벤트인 경우에는 폐기된다.

응용 프로그램 파트는 시스템 사용자 혹은 네트워크로부터 전송된 사용자의 입력을 프레임워크 파트로부터 전달받아, 응용프로그램 내부에서 자체적으로 처리해야 하는 이벤트를 처리하게 된다. 사용자의 입력에 따른 응용 프로그램의 화면 변화나, 멀티미디어 재생이 이 부분에 해당한다. 응용 프로그램에서 이벤트에 맞게 적절한 처리를 하고, 해당 이벤트를 다시 프레임워크 파트로 전송하게 된다. 그러므로 프레임워크 파트에서는 프레임워크에 이미 정의된 이벤트뿐만 아니라, 응용 프로그램에서 처리된 이후의 이벤트도 다시 처리하게 된다.

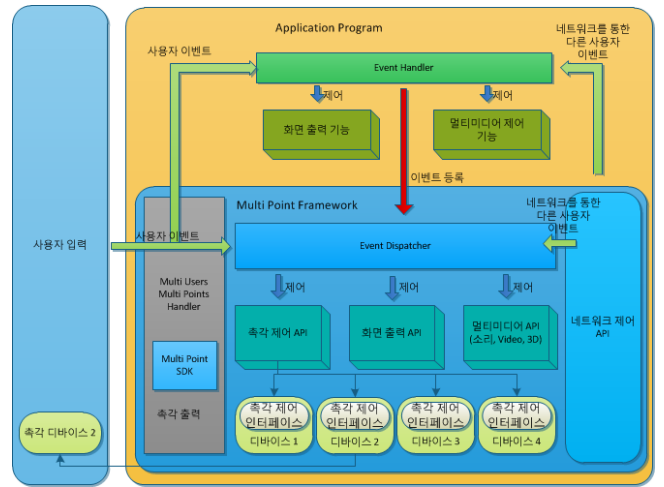
2.1 프레임 워크 파트

촉각 피드백을 지원하는 비접촉식 멀티포인트 프레임워크는 크게 4가지로 처리부로 구성되어있다.

- 1) 멀티 디바이스 처리부
- 2) 이벤트 처리 및 동기화 처리부
- 3) 촉각 피드백 처리부
- 4) 네트워크 제어부

비접촉식 멀티 포인트 프레임워크는 앞에서 언급한 것과 같이 다수의 사용자가 다수의 디바이스를 이용하여 시스템을 제어하는 것을 기본으로 한다. 윈도우즈 계열의

운영체제(Windows Family OS)에서는 기본적으로, 다수의 몇 몇의 모델에서만, 디바이스를 연결했을 때, 각자의 포인터를 가질 수 있다. 하지만, 대부분의 일반적인 사용자 인터페이스 디바이스 프로토콜(Human Interface Device Protocol, HID Protocol)을 가진 디바이스를 이용하는 경우, 기본적으로 윈도우즈 계열의 운영체제에서는 한 개의 마우스 포인터만을 가질 수 있게 되어있다. 즉, 두 개의 마우스를 사용하는 경우, 화면상에는 한 개의 마우스 포인터만이 움직이게 되는 것이다. 이러한 환경에서 다수의 사용자가 이용하게 되는 멀티포인트는 한계가 있으며, 이러한 한계점을 극복하기 위한 처리부가 멀티 디바이스 처리부이다.



(그림 2) 촉각 피드백 기반의 멀티 포인트 프레임워크 구조도

멀티 디바이스 처리부에서는 시스템이 구동하면, 시스템에 접속된 디바이스를 찾아내고, 피드백을 제공하기 위해 접속된 디바이스에 각각 아이디를 부여한다. HID Protocol을 통해서 들어오는 각 디바이스의 위치 정보를 처리해서 화면에 출력하고 디바이스 이벤트(좌클릭, 우클릭, 포인터이동 등)를 획득한다. 멀티 디바이스 처리부에서 포함되어 있는 Multi Point SDK는 디바이스 정보를 획득하여, 이러한, 접속된 각 디바이스에 아이디를 부여하고, 각 디바이스마다 발생하는 이벤트를 획득하는 역할을 하게 된다. 디바이스 이벤트를 처리한 후에 발생하는 시/촉각 피드백을 위해서 디바이스 이벤트, 디바이스 아이디를 이벤트 처리부에 전송한다.

이벤트 처리 및 동기화 처리부에서는 사용자 입력에 의해서 발생한 디바이스 이벤트를 처리하고, 각 디바이스에 대한 동기화 여부 및 이벤트 동기화를 처리한다. 사용자로부터 획득된 입력정보는 이벤트 처리부를 통해서 촉각 제어부, 화면 출력부, 멀티미디어 출력부로 전송된다. 이때, 각각의 처리부가 순차적으로 발생하는 것을 방지하기 위해서, 촉각 제어부, 화면 출력부, 멀티미디어 출력부에 이벤트 데이터를 같이 전송시킨다. 각각의 처리부에서는

사용자의 이벤트에 맞게 촉각 피드백을 제시하기 위한 처리를 한다던가, 화면상의 포인팅을 이동한다던가, 멀티미디어 출력(사운드 출력, 비디오 재생 등, 화면 출력을 제외한 멀티미디어 출력)을 위한 처리를 동시에 시작한다. 이러한 동시성 문제를 처리하기 위해서 동기화 처리부가 멀티 프로세싱을 통해서 한 번에 처리해 나간다.

촉각 피드백 처리부에서는 사용자의 디바이스 이벤트 정보와, 디바이스 아이디를 이용하여, 어느 디바이스에 어떤 촉각 정보를 제공할지를 결정한다. 촉각은 인체에서 시각, 청각, 촉각 중에서, 1ms안에 피드백을 제시해야 하는 예민한 부분에 해당하므로, 촉각 출력의 지연 현상이 발생할 경우, 사용자가 인지하기 쉽고, 불편함을 느낄 수 있게 된다. 따라서 촉각 피드백 처리부에서는 다른 제어 처리보다 빠른 처리가 될 수 있도록 처리 우선순위를 가장 높게 지정하여야 한다. 또한, 촉각 피드백 처리부는 촉각 제어 인터페이스에 연결되어 직접적으로 디바이스에 데이터를 전송하는 제어부 역할을 담당하게 된다. 그러므로 다수의 디바이스를 처리하기 위해서 디바이스 별로 촉각 피드백을 위한 데이터 전송을 병렬로 처리해야 디바이스 전송 오류에 따른 타 디바이스로의 오류 전파나 지연 현상을 최소화할 수 있다.

네트워크 처리부에서는 네트워크를 통해서 전달되는 외부 지역에 있는 사용자의 디바이스 이벤트 정보를 획득하여, 이벤트 처리부로 전달하는 기능을 담당한다. 외부로부터 획득된 정보는 이벤트 처리부를 통해서 각 이벤트를 처리하게 된다. 뿐만 아니라, 외부 사용자의 네트워크 상태 체크 및 접속, 복구, 유지를 담당하여, 안정적인 네트워크 서비스를 제공하도록 구성된다. 뿐만 아니라, 네트워크를 통해서 전송된 외부 사용자의 디바이스 아이디를 생성하고, 디바이스 이벤트 정보를 이용하여, 가상 디바이스가 접속된 상태를 만들어 낸다. 또한, 사용자 간의 상호작용에 의한 피드백생성을 위해서, 타 사용자의 이벤트에 의한 피드백 전송도 이루어진다. 따라서 이벤트 처리 및 동기화 처리부는 네트워크 처리부나 멀티 디바이스 처리부의 디바이스에 상관하지 않고, 이벤트를 처리해 나간다.

네트워크 패킷은 IP 정보, 디바이스 아이디 정보, 디바이스 X, Y, Z 위치 정보, 이벤트 타입정보를 묶어서 보내게 된다. 디바이스 아이디는 프로그램 구동 시에 부여하지만, 네트워크 접속 시에 업데이트를 함으로써, 아이디 중복을 해결할 수 있다.

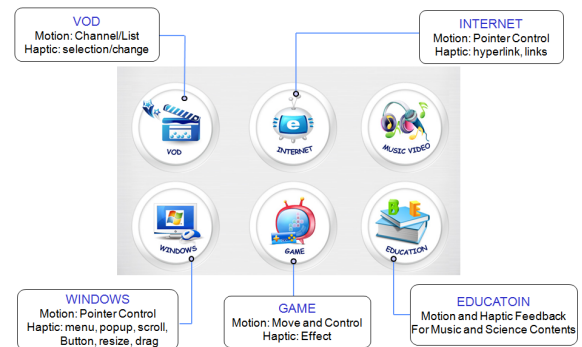
2.2 응용 프로그램 파트

촉각 피드백을 지원하는 비접촉식 멀티포인팅 프레임워크에서 응용 프로그램 파트는 프레임워크 파트를 제외한 부분이다. 응용 프로그램 파트는 응용 프로그램에 종속적인 부분을 처리하게 된다. 프레임워크가 응용 프로그램과 상관없이 동일한 부분이라면, 응용 프로그램 파트는 응용 프로그램에 따른 독립적인 부분이다. 응용 프로그램 파트에서는 프레임워크에서 전달받은 디바이스 이벤트를 자체

적으로 처리하는 이벤트 처리부가 따로 존재하며, 이 이벤트 처리부도 프레임워크 내부의 이벤트 처리부와 마찬가지로, 네트워크를 통해서 들어오는 이벤트와, 사용자로부터 전달되는 이벤트를 전달받아 처리하는 역할을 한다. 이 응용 프로그램 파트의 이벤트 처리부는 전달받은 이벤트를 처리하고, 추가적으로 다시 프레임 워크에서 처리해야 할 이벤트가 생기는 경우, 해당 이벤트를 프레임워크 내부로 전송한다.

3. 프레임워크의 응용

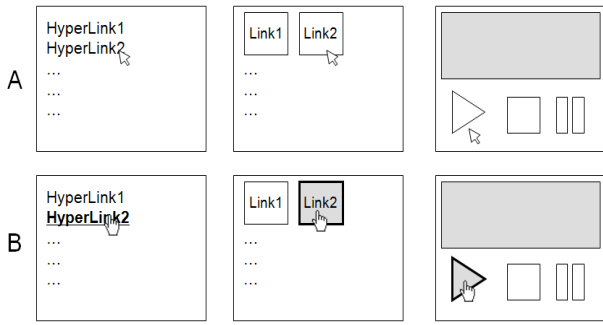
촉각 피드백을 지원하는 비접촉식 멀티포인팅 프레임워크를 이용하여 (그림 3), (그림 4)와 같은 응용 프로그램을 제작하였다. (그림 3)은 핸드헬드(Handheld)형 촉각 디바이스[5]를 이용하여, 메뉴를 조작하고, 메뉴 조작에 따른 촉각 피드백을 얻을 수 있도록 제작하였다. (그림 4)는 다수의 촉각 디바이스를 이용하여, 각 사용자가 문제를 풀 수 있도록 하는 응용 프로그램을 제작하였다. 포인터가 메뉴나 링크, 답안에 올라가거나, 메뉴에서 빠져나갈 때, 클릭할 때 등에 촉각 정보를 사용하여, 사용자가 좀 더 쉽고 빠른 조작을 할 수 있도록 제작되었다. (그림 5)는 (그림 3), (그림 4)에서 사용한 메뉴 조작 시, 상태 A에서 상태 B로 이동할 때 촉각 피드백을 제시하는 예시를 나타낸 것이다.



(그림 3) 촉각 피드백 장치를 이용한 IPTV User Interface 응용 프로그램



(그림 4) 촉각 피드백 장치를 이용한 멀티 포인팅 응용 프로그램의 예시



(그림 5) 메뉴 사용에 따른 촉각 피드백 제시 예. 링크에 포인터 오버되는 경우에 촉각 피드백을 제공.

4. 결론

본 논문에서는 촉각 피드백을 지원하는 비접촉식 멀티-포인팅 프레임워크를 제안하였다. 이를 위해서, 촉각 피드백을 지원하고, 멀티 포인팅 및 네트워크를 제공하기 위한 구조 설계하였으며, 프레임워크를 이용하여, 구현된 실제 시스템을 소개하였다. 멀티포인팅 기술은 다수의 사용자가 협업을 하는데 유용한 기술이며, 여러 사용자를 한 시스템에서 처리하는 기술이므로, 많은 하드웨어적인 리소스를 절약할 수 있게 해주는 기술이다. 뿐만 아니라, 국/내외에서는 아직 크게 연구되지 않은 미개척 분야이므로, 많은 시도 및 도전이 필요하다. 현재로서는 네트워크 기능을 제외한 프레임워크 기능이 구현된 상태이며, 추후 연구로는 현재의 멀티-포인팅 프레임워크를 네트워크상에서 안정적이고 범용적으로 사용할 수 있도록, 구현할 계획이다.

Acknowledgement

본 연구는 지식경제부 및 한국산업기술평가관리원의 IT 신성장동력핵심기술개발 사업의 일환으로 수행하였음. [KI002096, 비접촉식 멀티포인트 실감 인터랙션 기술 개발]

참고문헌

[1] B.Buxton. "Multi-Touch Systems that I Have Known and Loved.", <http://www.billbuxton.com/multitouchOverview.html>

[2] Iphone, Apple. Inc, <http://www.apple.com/iphone>

[3] Nimish Mehta, "A flexible human machine interface.", 1982.

[4] Microsoft Surface, <http://www.microsoft.com/surface/>

[5] K.U. Kyung, J.Y Lee, M.A. Srinivasan, "Precise Manipulation of GUI on a Touch Screen with Haptic Cues," Proceedings of the World Haptics 2009, pp.202-207, IEEE Computer Society, 2009

[6] A.D. Wilson, "TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction", ICMI'04, October 13.15, 2004.

[7] Q.Yan, Y.Wu, Z.Liu, H.Zhang, "Mimosa: A

Multi-touch System Used for Virtual Exhibition in Public Spaces", 2010 2nd International Conference on Industrial and Information Systems

[8] U.Laufs, C.Ruff, J.Zibuschka, "MT4j - A Cross-platform Multi-touch Development Framework", Workshop: Engineering patterns for multi-touch interfaces, 2010, p. 52-57

[9] C.Scholliers, L.Hoste, B.Signer, W.D.Meuter, "Midas: A Declarative Multi-Touch Interaction Framework", TEI'11, January 22.26, 2011

[10] D.Kammer, M.Keck, "Taxonomy and Overview of Multi-touch Frameworks: Architecture, Scope and Features"

[11] Microsoft, "Multipoint Mouse SDK", <http://www.microsoft.com/multipoint/mouse-sdk/>

[12] Multipoint TUIO, <http://multipointtuo.codeplex.com/>