

유비쿼터스 환경에서 사용자 동작을 인식하기 위한 입력장치

김종우, 김정래, 전수빈, 박충명, 정인범*
강원대학교 컴퓨터정보통신공학전공
e-mail:jw_kim@snslab.kangwon.ac.kr

An Input Device to cognize the Human Motion in Ubiquitous Environment

Jong-Woo Kim, JeongRae Kim, Soo Bin Jeon, Chongmyung Park, In-Bum Jung
Program of Computer and Communications Engineering,
Kangwon National University

요 약

기존에 사용 중인 컴퓨터 입력기기인 키보드와 마우스는 유비쿼터스 환경에서의 입력기기로서는 휴대성과 직관성의 결여로 사용하기에 부적합하다. 유비쿼터스 환경에서는 휴대성과 직관성이 강화된 차별화된 입력기기가 필요하다. 본 연구에서는 유비쿼터스 환경에 맞는 입력기기에 사용 가능한 센서에 대해 실험하고 그 결과를 바탕으로 유비쿼터스 환경에 맞는 입력기기를 제안하고자 한다.

1. 서론

유비쿼터스 환경에서 급격히 증가하는 컴퓨터 시스템의 효율적인 사용을 위해 직관적인 인터페이스의 개발이 필요하다. 기존의 마우스는 데스크탑 컴퓨터의 입력장치로는 적합하지만 유비쿼터스 환경에서 사용하기에는 직관적이지 않고 거리에 제약이 있어 사용에 불편함을 준다. 따라서 이러한 문제를 해결하기 위한 많은 연구가 이루어지고 있다[1].

본 연구에서는 사용자와 컴퓨터의 효율적인 상호작용을 위해 여러 가지 센서를 실험하고 데이터를 분석하였고 이를 바탕으로 사용자 동작 인지형 장치의 하나인 장갑형 마우스를 제작하였다. 본 논문의 2장에서는 관련연구인 사용자 움직임 추적기법과 센서 네트워크에 대해서 설명하고, 3장에서는 입력기기의 설계 및 구현에 대해서 설명한다. 4장에서는 결론을 맺고 본 논문을 마무리한다.

2. 관련연구

2.1 사용자 움직임 추적기법

사용자의 움직임을 추적하는 방법은 크게 두가지로 나누어 볼 수 있는데 Vision Tracking 기법과 센서 하드웨어를 이용한 기법이다.

Vision Tracking 기법은 카메라를 통하여 움직임 정보나 위치를 영상 샘플링과정을 통해 직접적으로 추출해 낼 수 있다는 장점이 있다. 하지만 장치의 사용을 위해서는 카메라의 적절한 위치 및 초점, 광량의 초기 세팅이 까다

롭고 사용자가 항상 카메라의 앞에 위치해야 한다는 단점이 있다. 또한 Vision Tracking 기법은 화면의 포착에서 샘플링 처리까지의 시간 지연으로 인해 높은 주기의 움직임 측정에는 부적합하다.

사용자 움직임 추적의 다른 방법으로는 관성 센서나 가속 센서등의 센서 하드웨어를 이용한 기법이 있다. 이 기법은 센서의 노이즈 문제점이 존재하지만 Vision Tracking 기법보다 작은 움직임을 포착하기에 적합하고 지정된 위치가 아닌 센서 통신범위 어디서라도 작동이 가능하다는 장점이 있다[2,3].

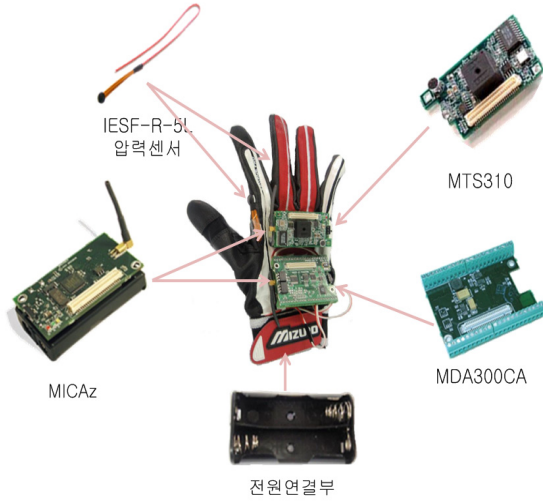
2.2 센서 네트워크

센서 네트워크란 센서를 통하여 사물의 인식정보를 기본으로 주변의 환경정보를 탐지하여 이를 실시간으로 네트워크에 연결하여 생성된 정보들을 관리하는 통신망이다. 센서 네트워크는 그 성향이 비슷한 애드혹(Ad-hoc) 네트워크에 비해 상대적으로 규모가 커 노드수가 10~100,000 정도가 되어 센서 노드들이 조밀하게 배치될 수 있다. 그리고 센서 네트워크에서는 브로드캐스팅을 주로 이용한다.

노드의 위치는 필요에 따라 유동적이고 노드 당 가격이 저렴하므로 대규모 센서 네트워크의 구현이 용이해야 한다. 센서 네트워크의 수명은 각 노드의 배터리에 의존하므로 노드의 전력 소비량이 최소가 되어야 하며 일부 노드에 결함이 발생하여도 다른 경로를 통하여 네트워크 기능을 유지해야한다. 또한 센서 노드들의 전원, 메모리 등의 하드웨어 자원뿐만 아니라 계산능력 또한 매우 제한된다. 거기에 더하여 센서 노드들은 하나의 목적물을 위하여 배치된 수가 많기 때문에 IP 주소와 같은 국제적인 식별자를 갖는 것이 어렵다[4].

- 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신
인력 양성사업으로 수행된 연구 결과임.

*교신저자



(그림 1) 사용자 동작 인지형 장치의 구성

3. 사용자 동작 인지형 장치의 설계 및 구현

3.1 시스템 설계

본 연구에서는 유비쿼터스 환경에서 사용자의 움직임을 추적하기 위한 입력기와 TinyOS기반 애플리케이션을 개발하였다. 실험은 MTS310과 MDA300CA, IESF-R-5L 압력센서를 이용하여 가속도 및 버튼 입력을 측정하고 분석하였다[5,6].

분석된 데이터를 바탕으로 MTS310의 2축 가속도 센서와 압력센서 2개를 활용하여 손목의 기울기 값으로 움직임을 제어하는 장갑형 마우스를 구현하였다.

사용자 동작 인지형 장치는 그림 1에서 보는 것과 같이 2개의 MICAz 모트와 하나의 MTS310 센싱보드, 두 개의 IESF-R-5L 압력센서, 하나의 MDA300CA 인터페이스 보드를 이용하여 사용자 동작 인지형 장치로 장갑형 마우스를 개발하였다. 사용자 동작 인지형 장치로부터 데이터를 수신하여 컴퓨터로 전송하는 부분에는 프로그래밍 보드와 MICAz 모트를 사용하였다.

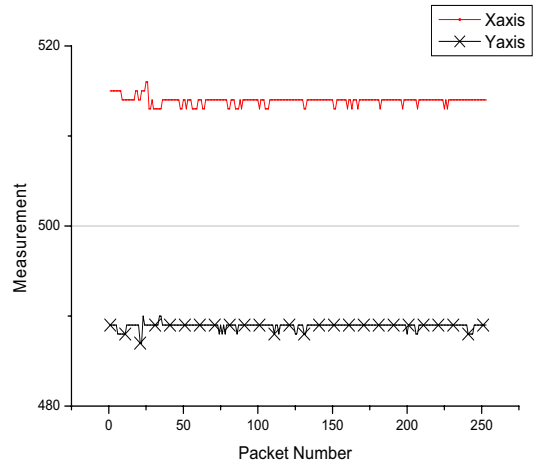
두 개의 압력센서는 MDA300CA 인터페이스 보드에 연결되어 동작하고, MTS310 센싱보드와 MDA300CA 인터페이스 보드는 각각 하나의 MICAz 모트에 부착되어 센싱된 정보를 전송한다. 각각의 센싱된 정보들은 4개의 채널을 사용하여 패킷단위로 전송되며 한 패킷에는 10개의 데이터가 저장되어 전송된다.

압력센서 하나는 버튼의 입력값으로 사용되었고, 다른 하나는 2축 가속도 센서의 데이터를 컴퓨터에서 마우스의 움직임에 이용하거나 버리는 기능키 역할로 사용하였다. 기능키의 사용으로 전원을 끄지 않고도 마우스의 움직임을 제어 할 수 있게 되었다.

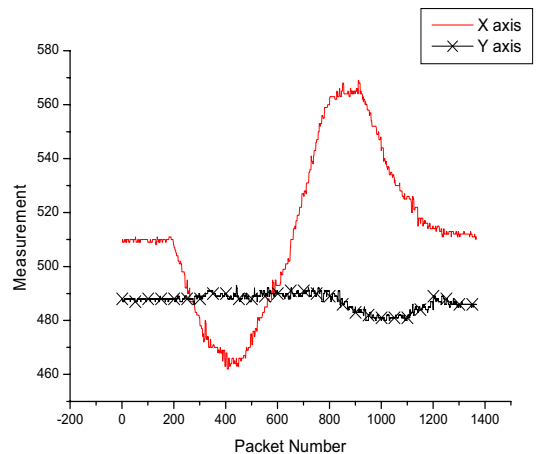
3.2 가속도 센서를 이용한 움직임 추적 실험

사용자의 움직임을 추적하여 컴퓨터 시스템을 제어하기 위해 사용자의 손동작 정보를 파악 할 필요가 있다. 사용자의 손동작 정보를 파악하기 위하여 MTS310의 2축 가속도 센서를 이용해서 실험하였다. MTS310 2축 가속도 센서를 손에 부착하고 평상시 데이터와 좌/우, 상/하로 움직였을 때의 데이터를 수집하였다.

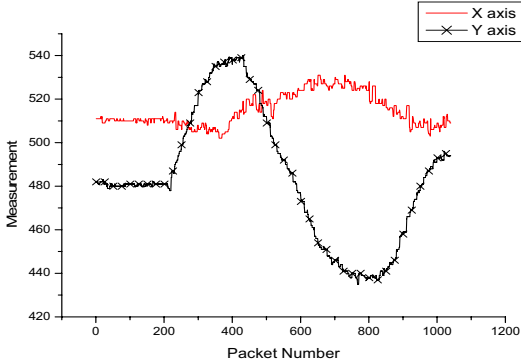
그림 2, 3, 4의 그래프를 이용하여 Threshold 값을 구하고, 이를 이용하여 마우스 포인터의 이동정보를 해석하였다. 그림 2에서와 같이 아무런 움직임이 없는 상태의 x좌표의 기본 값은 평균 514의 값을 가지고, y좌표는 489의 값을 가진다. 이 값들을 기준으로 그림 3, 4의 데이터를 정규화 하면 그림 3, 4으로 나타나는 값들이 0값을 기준으로 상측과 우측으로의 움직임은 +값이 되고, 하측과 좌측으로의 움직임은 -값으로 나타난다.



(그림 2) 움직임이 없을 때의 데이터



(그림 3) 좌/우의 기울기에 따른 데이터



(그림 4) 상/하의 기울기에 따른 데이터

```

01 if(readBuffer[10] == 0x07)
02 {
03     int y = ((readBuffer[13] & 0x000000ff)<<8) |
(readBuffer[12] & 0x000000ff) ;
04     y = y - 489;
05     p.y -= y;
06     if( y >= 8 || y <= -8 )
07     {
08         SetCursorPos(p.x, p.y);
09     }
10 }
11 else if(readBuffer[10] == 0x09)
12 {
13     int x = ((readBuffer[13] & 0x000000ff)<<8) |
(readBuffer[12] & 0x000000ff) ;
14     x = x - 514;
15     p.x += x;
16     if( x >= 8 || x <= -8)
17     {
18         SetCursorPos(p.x, p.y);
19     }
20 }
    
```

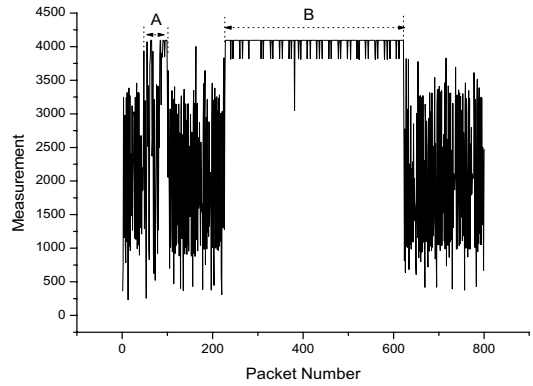
(그림 5) 마우스 좌표변환 알고리즘

그림 5는 이러한 실험결과를 바탕으로 이동정보를 해석하는 알고리즘이다. 알고리즘 내에서 처리되는 데이터는 패킷단위로 전송되어 처리되기 때문에 패킷마다 어떤 데이터의 패킷인지의 구분이 필요하다. 그림 5의 알고리즘에서 01번과 11번 라인의 if문으로 비교하는 부분이 각각 x, y 데이터인지를 판단하는 부분이다. 03번 13번 라인은 패킷에서 처음 나오는 데이터부분으로 빅엔디안 형식으로 된 2bytes 데이터를 리틀엔디안 형식으로 변환한 후 계산해야 한다. 변환한 데이터를 Threshold 값을 통해 정규화한 값에서 손목의 미세한 움직임이나 센서에서 발생하는 노이즈 값의 범위 이외의 값을 마우스 포인터 값 변경에 사용한다. 알고리즘 상의 변수 p는 마우스의 x, y 좌표를 가지고 있는 구조체 변수이다.

3.3 압력 센서를 이용한 마우스 클릭

사용자의 움직임을 추적하여 컴퓨터 시스템을 제어하기 위한 명령의 실행과 클릭을 구현하기 위해 버튼으로 사용할 압력센서를 실험하였다[7].

그림 6에서 버튼으로 사용하는 압력센서가 압력을 감지한 경우 그림 6의 구간 B와 같이 4095의 값이 발생하고, 노이즈 값으로 인해 3821까지 변화 한다는 것을 알 수 있다. 구간 A는 노이즈 발생 구간이다. 그림 7은 노이즈 때문에 발생할 수 있는 오동작을 방지하기 위한 알고리즘이다. 01번 라인에서 버튼 데이터 관련 패킷인지를 판별하고 04번 라인의 for문을 통해 3821이상의 값이 한 패킷에 일정 개수 이상 들어오는 지를 판별하여 마우스 클릭 이벤트를 발생 시켰다. 노이즈로 발생한 값과 압력센서에 압력을 가해 발생한 값의 차이를 구분 할 수 없기 때문에 연속적으로 값이 발생하는 경우에만 해당 명령을 실행하였다. 2축 가속도 센서를 제어하는 압력센서의 패킷에서 3821이상의 값이 6개 이상 확인되면 true/false값을 변경해 주었다.



(그림 6) 압력감지 데이터

```

01 if(readBuffer[10] == 0x04)
02 {
03     button_check = 0;
04     for (int j = 12 ; j< 33 ; j = j+2)
05     {
06         if((((readBuffer[j+1] & 0x000000ff)<<8)
| (readBuffer[j] & 0x000000ff))>= 0x0eed)
07         {
08             button_check++;
09         }
10     }
11     if(button_check >=6)
12     {
13         mouse_event(MOUSEEVENTF_LEFTDOWN,
p.x, p.y, 0, 0);
14         mouse_event(MOUSEEVENTF_LEFTTUP,
p.x, p.y, 0, 0);
15     }
16 }
    
```

(그림 7) 버튼 클릭 감지 알고리즘

4. 결론 및 향후계획

유비쿼터스 환경에서 직관적이고 효율적인 입력 인터페이스의 제작을 위해 2축 가속도 센서와 압력센서를 사용하여 장갑형 마우스를 개발하였다. 개발된 장갑형 마우스를 이용하여 프리젠테이션과 게임, 웹서핑 등의 실질적인 동작을 제어하였다. 프리젠테이션에서는 손에 특정한 도구를 들고 있지 않고도 프리젠테이션이 가능한 편리함을 느낄 수 있었고, 게임에서는 직관적인 움직임으로 인해 높은 몰입감과 체험감을 느낄 수 있었다.

본 연구에서 제작된 장갑형 마우스는 마우스 포인터의 움직임과 클릭과 더블클릭의 기능만이 지원되고 있다. 향후에는 다른 여러 센서들을 추가하여 직관적이고 사용이 편리한 차세대 사용자 동작 인식 입력기기의 연구를 진행할 계획이다.

참고문헌

- [1] 김창석, 구홍서, 이상범, 박태웅, 이성국, “유비쿼터스 컴퓨팅 기술과 인프라 그리고 전망”, 한국정보처리학회지 104, 7. 2003.
- [2] S. You, U. Neumann, “Fusion of Vision and Gyro Tracking for Robust Augmented Reality Registration”, IEEE conference on Virtual Reality, pp. 71 - 78, 2001.
- [3] 김효승, 박광규, 박현구, 차호정, “MGlove: 센서 네트워크 환경에서의 사용자 센싱에 기반한 입력 장치”, 한국정보과학회 31권 2호, 10. 2004.
- [4] D. Culler, D. Estrin, M. Srivastava “Overview of Sensor Networks” IEEE Computer Society pp.41-49 August 2004.
- [5] TinyOS (<http://www.tinyos.net/>)
- [6] Crossbow (<http://www.xbow.com/>)
- [7] IESF-R-5L 압력센서 (<http://www.cui.com/>)