

차량 내 제스처 인식을 위한 밴드형 웨어러블 디바이스 개발 및 서비스에 관한 연구

박인혜, 이상엽, 고재진
전자부품연구원

요약

본고에서는 차량 내 운전자 제스처 인식을 위한 웨어러블 디바이스 플랫폼과 차량 시스템 모델을 소개한다. 특히 저가형 가속도, 자이로스코프 센서를 적용한 웨어러블 디바이스용 HW를 설계하고, 제스처 인식률을 높이기 위한 SW 시스템 및 알고리즘에 대해 설명한다. 운전 중 가장 많이 제어하는 편의 기능을 기준으로 제스처를 구분하고, 제스처 마다 발생하는 센서 값의 특징을 이용해 잡음에 강한 제스처 인식 시스템에 대해 설명한다. 마지막에 진행한 본 논문의 실험을 통해 본 고에서 설명한 제스처 인식 시스템이 잡음과 방향성을 갖는 동작인식에 강함을 확인할 수 있다.

I. 서론 및 개요

최근 차량 내 운전자의 제스처를 통한 다양한 인터랙션 기술이 선보이고 있다. 운전자는 간단한 제스처를 통해 차량 내 멀티미디어 시스템 제어, 공조 시스템 제어 등 안전운전이 가능한 범위에서 편리한 기능을 사용하고 있다. 최근 BMW에서는 모션인식 카메라를 이용한 운전자 제스처 인식을 통해, 차량 내 헤드유닛 UI의 방향전환, 오디오 볼륨 조절, 전화 수신 등 다양한 제스처 응용기술을 적용하고 있다. 이해를 돕기 위해 다음



그림 1. 카메라기반 제스처 인식 시스템(출처: BMW 홈페이지)

〈그림 1〉에 BMW에서 적용한 카메라기반 제스처 인식 시스템 동작 예를 나타냈다.

카메라기반 제스처 인식 기술은 개발 비용이 높고, 외부 조건에 의한 빛반사, 빛번짐 현상으로 인해 카메라가 인식오류를 일으킬 수 있다는 단점이 있다. 또한 제스처를 수행하는 영역이 지정되어 사용자가 반드시 지정 영역에서 움직여야 한다는 단점이 있다. 이러한 카메라기반 제스처 인식 방법의 단점을 보완한 웨어러블 디바이스에서의 제스처 인식 방법이 고려된다. 차량에서 진행되는 제스처 인식 환경의 특성상, 사용자가 수행한 동작은 즉시 실행되어야 하는 조건이 있다. 또한 사용자의 주의 분산 없이 제스처 위치에 관계없이 동작이 인식되어야 응용 프로그램을 사용 할 수 있어야 한다. 따라서, 본 연구에서는 사용자가 착용하고 있는 웨어러블 디바이스를 이용하여 빠르고 정확하게 차량 내 제스처 인식을 수행하고, 인식 위치에 관계없이 응용 프로그램을 제어할 수 있는 시스템을 설계하고 이에 대해 설명한다.

웨어러블 디바이스를 이용한 차량 인포테인먼트의 조작성이 편리한 방법임에도 불구하고 차량 내 환경에서 운전자의 제스처를 인식하는 것은 많은 문제를 안고 있다. 잦은 진동과 움직임 때문에 정확한 제스처 인식이 어려운 것이 가장 크다. 따라서 운전 중의 제스처 동작의 오류 저감을 위하여 몇 가지를 정의한다. 운전자의 제스처를 잘못 인식하는 오류 상태에 대해, 운전자 모션에 따라 생성되는 센서 값 내부 오차, 사용자의 제스처를 인식하는 디바이스의 데이터 처리오류, 차량의 이동 환경 중에 발생하는 외부 장애물에 대한 오류를 정의한다.

다양한 동작인식 방법들이 많은 측면에서 연구와 발전을 거듭하고 있다. 연구적인 측면에서는 웨어러블 디바이스에 삽입된 저가형 센서인 MEMS(micro inertial measurement unit)를 이용하여 제스처를 인식하는 방법이 많이 소개됐다. [2]에서는 센서들 값을 이용해 HMM(hidden Markov model)으로 동작을 유추했다. 장착된 센서 값의 크고 작음에 따라 총 7가지 동작을 구분할 수 있도록 했다. [3]에서는 휴대폰에 삽입된 MEMS 센서에서 발생하는 값으로 방향을 유추하여 로봇에게 명령을 전달하는 연구를 진행했다. 발생하는 센서 값으로 로봇의 방향을

전달하는 수준으로 제스처 인식 범위를 지정했다. [4]는 앨런 분산(Allen variance)을 이용해 웨어러블 장치에서 발생하는 센서 데이터의 의미, 상태를 구분하고자 했다. 주로 걷기와 계단 오르기 등의 특정한 동작을 대상으로 주체의 동작을 구분했다. 이러한 연구들은 모두 잡음이 없는 평지에서 제스처 인식이 고려됐기 때문에 움직임이 있는 환경에서는 그 정확도를 기대하기 힘들다.

근전도 신호를 이용한 제스처 인식에 관한 연구도 진행됐다 [5][6]. MEMS와 달리 동작의 속도나 방향보다 손가락 모양, 손의 동작으로 인해 발생하는 근전도 센서에서는 다른 방향으로의 제스처 인식 연구가 진행됐다. [5]에서는 자기회귀 이동평균 모델을 제스처 인식에 사용했다. 손가락 동작과 손의 모양에 따라서 변화되는 팔의 근육 센서 값에 따라서 패턴 인식으로 제스처를 구분했다. 또한 [6]에서는 다양한 동작을 먼저 모델링하여 동작에 매칭되는 채널 값에 따라서 제스처 인식을 진행했다. 이 근전도 센서는 반드시 팔에 밀착돼서 부착돼야 동작된다는 센서 특성적인 단점을 지닌다.

이렇듯 제스처 인식에 관련된 선행 연구들이 존재한다. 그러나 차량내 환경에서의 제스처 인식 알고리즘은 거의 없다. 차량에서는 움직임이 많고 잔 진동으로 인해 센서 오류가 많이 발생하여 정확한 제스처 인식이 어렵기 때문이다[7][8][9]. 본 논문에서는 저가형 장치를 타겟으로 오류를 최소화할 수 있는 웨어러블 디바이스 플랫폼 모델을 정의하고 오류 저감에 대한 성능을 평가한다. 웨어러블 디바이스에 적용할 수 있는 플랫폼을 이용해 차량 내 운전자의 제스처에 대한 오류를 저감 할 수 있는 알고리즘을 함께 제안한다. 제안하는 알고리즘은 센서 값 독립적인 특성을 이용해 패턴을 분석하지 않고, 센서의 모든 축에서 발생한 데이터 간의 상호 상관관계를 이용해 제스처 인식에 사용한다. 차량에서 발생하는 작은 움직임과 잔 진동은 상호 상관관계 계산 중 상쇄되므로 보다 높은 인식률로 제스처를 구분해 낼 수 있다.

표 1. 제스처 별 차량 내 동작의 예

운전자 제스처	차량 동작
왼쪽 비틀기	이전 음악 재생
오른쪽 비틀기	다음 음악 재생
위로 굽히기	오디오 전원 켜기
아래로 굽히기	오디오 전원 끄기
시계방향 회전	오디오 음량 키우기
반시계방향 회전	오디오 음량 줄이기

본 연구에서는 차량 내 운전자가 가장 많이 제어하는 오디오 시스템을 기반으로 <표 1>과 같은 제스처 예시를 통해 성능을 평가하였다. 성능 평가를 위해서 차량 외부 환경에 급제동, 급출발, 방지턱 변화 등에 대해서는 제스처 실행 시 오류로 인식하고 해당 데이터를 제외하고 재실행하는 방식을 사용하였다.

본 논문의 구성은 다음과 같다. 먼저 웨어러블 디바이스 HW 플랫폼에 대해서 설명한다. 전체 시스템 구조와 HW 플랫폼의 특성, 생성하는 센서 데이터의 프레임 구조에 대해 설명한다. 다음으로 제스처 인식 시스템의 구성을 포함한 제스처 인식 알고리즘에 대해 설명한다. 다음으로 개발된 웨어러블 디바이스와 제안된 제스처 인식 알고리즘을 이용한 실험 환경과 그 결과에 대해 설명한다. 마지막으로 본 논문의 결론과 함께 향후 연구에 대해 설명한다.

II. 웨어러블 디바이스 플랫폼

1. 웨어러블 디바이스 시스템

웨어러블 디바이스 시스템 설명에 앞서, 사용자 움직임 정보를 발생시켜 차량으로 데이터를 전달하는 전체 시스템의 구조에 대해 설명한다. 본 논문에서 지정한 차량 내 웨어러블 디바이스와 작용자의 모션 정보를 전달하는 전체 구조를 <그림 2>에 나타냈다. 가장 먼저 사용자의 웨어러블 디바이스에 부착된 6축 센서가 움직임 데이터를 발생시킨다. 발생된 데이터는 블루투스(BLE: Bluetooth low energy)를 통해 차량 내 헤드유닛 시스템에 전달된다. 가공되지 않은 RAW데이터 형태인 움직임 데이터는 웨어러블 디바이스 내에서 내부 오류가 1차적으로 보정된다. 보정된 센싱 데이터는 헤드유닛 시스템에서 제스처 정

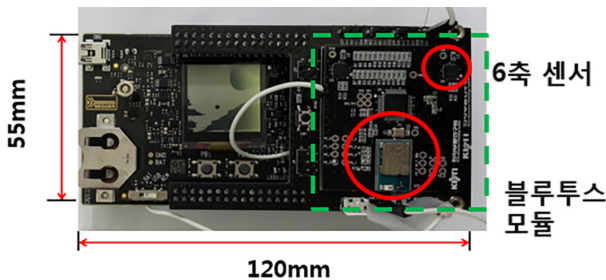


그림 2. 웨어러블 디바이스 시스템 구조

보로 변환된다. 변환 과정에서는 제안된 제스처 인식 알고리즘(3장에서 설명)이 사용된다. 헤드유닛 시스템에서는 제스처 인식 알고리즘을 적용할 때, 센서 축 별 독립적인 특성이 아닌 상호 상관관계를 이용한 구분으로 제스처 인식에 사용되는 오류의 비중을 줄이고 사용자의 움직임 데이터의 보다 정확한 해석을 돕는다.

2. 웨어러블 디바이스 플랫폼

움직임 정보를 발생시켜 차량으로 전달하는 웨어러블 디바이스의 HW 플랫폼을 <그림 3>에 나타냈다. <그림 3>에 나타나 있듯이, HW 플랫폼에 움직임 정보를 수집하는 6축 센서와 블루투스 모듈을 장착했다. MCU는 Cortex-M 시리즈의 EFM32 HG32를 사용하여 센서의 수집과 블루투스 통신 그리고 센서 값의 1차 가공이 가능하도록 했다. 밴드형 웨어러블 디바이스인 본 플랫폼은 초저전력을 목적으로 구현됐다. 따라서 내부 SW는 제스처 인식을 위해 센서 데이터 수집과 전달 등 최소한의 필요 동작만 실행하는 펌웨어로 구현됐다. <그림 3>의 HW 플랫폼에는 핵심 발생장치와 개발 및 전원 공급을 위한 레퍼런스 플랫폼이 함께 있다. 센서와 블루투스 모듈을 담은 핵심 보드는 <그림 3> 오른쪽 편에 점선으로 표시되었다..



Wearable Device Platform

그림 3. 웨어러블 디바이스 HW 플랫폼

적용된 6축 센서는 3축의 가속도 센서와 3축의 자이로스코프(Gyroscope, 이하 자이로) 센서이다. 두 센서는 발생하는 움직임 정보를 3차원의 가속도 값과 각속도 값으로 각각 표현한다. 저가형 웨어러블 디바이스에 적용된 이 센서는 비교적 정교한 센싱 값을 전달하지 못한다[7]. 따라서 사용자 센서 내부의 오차 값이 발생할 수 있으며, 센서 정보를 전달하는 데이터 손실에 대한 오류가 발생하고, 제스처 실행 상태에서의 외부 환경에 의한 오류가 발생하게 된다[10]. 이를 위해 개발된 HW 플랫폼의 내부에서는 발생하는 센서 모듈의 내부 오차를 저감을 위해 데이터 값의 1차 가공을 한다.

Start bit (2byte)	센싱 데이터 (6*n*2byte)	End bit (2byte)
----------------------	-----------------------	--------------------

그림 4. 센싱 데이터 전달 프레임 구조

3. 센싱 데이터 전달 프로토콜

웨어러블 디바이스의 센싱 데이터 해석을 위해서는 정형화된 데이터 전달 프로토콜이 필요하다. 전송 도중 발생하는 오류 존재 여부 및 센서 데이터의 전달 시작 시점을 파악해야 하기 때문이다. 다음 <그림 4>에 센싱 데이터 전달 프레임 구조를 나타냈다. 약속된 2byte의 start/end bit를 각각 센서 데이터의 앞과 뒤에 추가했다. 만일 데이터 전달시에 지정된 start/end bit와 값이 일치하지 않는다면, 본 프레임에 오류가 있으므로 판단되어 수집된 데이터 정보는 버려진다.

한 데이터 프레임에 담긴 센싱 데이터는 2byte 단위로 n묶음이 된다. 따라서 하나의 데이터 프레임의 크기는 4byte+(6축*n개*2)byte가 된다. 단, 하나의 데이터가 반드시 하나의 움직임을 의미하는 것은 아니며, 실시간으로 발생하는 1차 가공 센서 데이터를 포함한다.

잡음이 많은 환경에서는 제스처 인식을 위 사용자 움직임 시작 시점을 파악하는 장치가 필수적이다. 명확한 시점의 지정은 움직임 데이터와 무작위로 발생하는 잡음의 구분을 더욱 용이하게 한다. 따라서 개발된 웨어러블 디바이스에서는 센싱 데이터 전달 시에 초기화 구간을 삽입하여 제스처 인식을 위한 움직임 시작 시점을 지정해줬다. 이를 이용해 초기화 구간 직후의 센싱 데이터부터 움직임의 시작으로 인식토록 했다.

Ⅲ. 제스처 인식 알고리즘

1. 제스처 인식 시스템 구성

제스처 인식 시스템은 다음 <그림 5>에 나타난 것과 같이 사용자의 동작을 정확히 해석하는 것이 목적이다. 특히 잡음이 많은 차량 환경 상태에서도 정확한 제스처를 해석해 사용자가 필요로 하는 응용프로그램을 동작 시킬 수 있어야 한다. 따라서 본 고에서 제안한 시스템은 무작위로 발생하는 잡음과 무의미한 동작을 제거하기 위해 센서 값의 상관관계를 파악하여 이를 이용한 알고리즘을 적용한다.

제스처 인식 시스템 구조를 설명하기 위해 <그림 6>에 시스템에서 사용한 제스처 인식 시스템의 구성도를 나타낸다. <그림 3>의 구성도는 헤드유닛 시스템에서 동작하며, 모션인식을

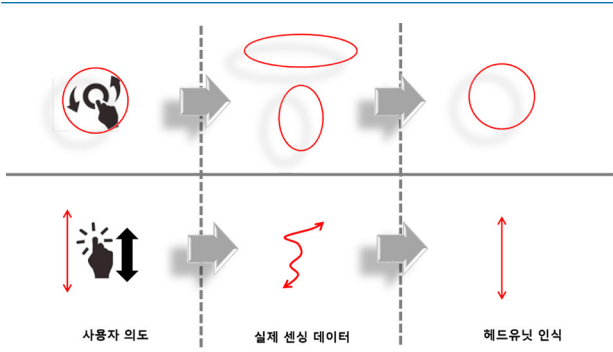


그림 5. 제스처 인식 시스템의 목적

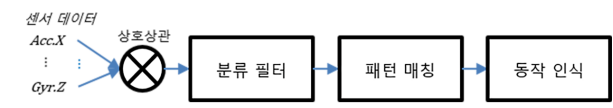


그림 6. 제스처 인식 시스템의 구성

위해 웨어러블 디바이스에서 발생한 센서 값의 상호 상관관계 (cross-correlation)를 계산한다. 그 다음으로, 얻어진 상관관계 정보를 이용해 무의미한 값을 필터링 하고 저장된 동작 DB 값과 비교하여 사용자의 제스처를 인식하도록 구성되었다.

제스처 인식 시스템의 구조를 포함한 웨어러블 스마트 플랫폼과 헤드유닛 시스템의 전체 동작 흐름도를 <그림 7>에 나타냈다. <그림 7>의 순서도에 ①과 ②로 크게 웨어러블 디바이스에서의 동작 부분, 차량 헤드유닛 시스템에서 센서 데이터 처리 부분으로 나누어있다. 특히 ②의 헤드유닛 시스템 데이터 처리 부분은 <그림 6>에 표시된 블록도로 순차적 처리된다. 헤드유닛 시스템 데이터 처리 부분은 총 세 번의 데이터 필터링이 일어난다. 분기의 순서대로 데이터의 무결성 검사, 유의미한 상호 상관관계 존재여부 검사, 제스처 DB일치여부 검사가 진행된다. 이 과정이 모두 통과되면 유의미한 제스처로 인식되어 응용프로그램의 실행이 가능해진다.

차량 내 적용 시스템 모델 구성도에 따라, 제안된 알고리즘은 웨어러블 디바이스로부터 전달받은 6축 센서 정보를 이용해 사용자의 제스처를 인식한다. 헤드유닛 시스템으로 전송된 데이터는 임베디드 HW의 특성상 낮은 샘플링 주기로 데이터를 양자화 하기 때문에 사용자의 제스처 정보를 상세히 표현 할 수는 없다. 따라서 정확한 움직임 정보를 표현하지 못하는 시스템 상에서 모든 축의 정보를 동시에 비교하고 해석하는 상호 상관관계 방법 적용이 적합하다. 또한 상호 상관관계 방법을 통해 모든 축에서 동시에 발생하는 잡음이 저감되기 때문에 임베디드 시스템 상에서 사용되는 다른 제스처 인식 장치보다 비교적 정확한 제스처 인식이 가능하다.

2. 데이터 선별과 제스처 인식

헤드유닛 시스템의 제스처 인식 시스템은 센서정보를 수신하면 가장 먼저 초기화 구간을 찾아 움직임 시작점을 체크한다. 움직임 시작점을 기준으로 센서 값 간의 상호 상관관계를 계산하여 유효한 움직임을 선별한다. 상호 상관관계는 0 값에 가까울 수록 서로 관련성이 적음을 의미하고 1 혹은 -1에 가까울수록 높은 상관관계를 가짐을 의미한다. 따라서 특정 수준 이하의 상호 상관관계 값은 유효한 움직임 데이터 구분에서 제외하고 제스처 인식에 사용한다. 이 방법은 제스처 동작 시에 모든 센서에서 동일한 패턴의 무작위 센서가 발생하기 때문에 센서 자체에서 발생한 무작위 잡음 값에 큰 의미가 없어진다. 따라서 잡음의 값이 무시되기 때문에 오류 저감의 효과를 동시에 취할 수 있다.

선별된 데이터 모음은 자동적으로 무의미한 동작을 걸러내는 작용을 한다. 특정 수준 이상의 선별된 상호 상관관계 값이 없거나, 미리 정의된 DB내에 일치하는 정보가 없는 경우는 무의미한 움직임으로 분류되어 삭제되기 때문이다. 수집된 데이터들은 미리 정의된 DB내의 상호 상관관계 정보와 비교되어 특정 제스처의 특성과 일치할 시에 해당 제스처로 분류된다. 즉, 수

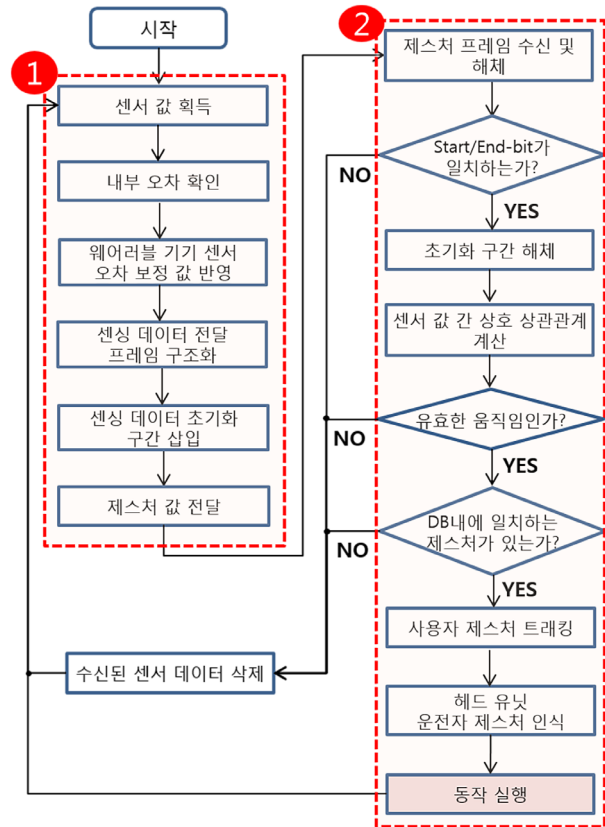


그림 7. 알고리즘 전체 동작 순서도

집된 상호 상관관계 값이 제스처 비교에 이용되는 것이 아니라, 수집된 값의 센서 관계가 제스처 비교에 이용된다. 이러한 방법은 복잡한 계산없이 빠르게 처리가 가능하다. 따라서 임베디드 시스템상에서 몇가지 제스처에 한하여 동작 구분을 빠르게 진행할 수 있다.

IV. 실험

1. 실험 환경 구성

웨어러블 디바이스와 제스처 인식 알고리즘이 적용된 시스템의 성능을 평가하기 위한 실험 환경 구성을 <그림 8>에 나타냈다. 웨어러블 디바이스 플랫폼과 블루투스를 통해 데이터를 수집하는 헤드유닛 시스템 통신부, 제스처를 인식할 수 있도록 데이터를 처리하는 헤드유닛 시스템 데이터 프로세싱부로 나뉜다.

차량용 헤드유닛 플랫폼 선정된 임베디드 HW 플랫폼은 현재 차량의 AVN에 많이 사용되는 Freescale사의 i.MX6 AI(auto-motive infotainment) 어플리케이션 보드를 채택했다. 이 플랫폼은 쿼드코어 Cortex-A9 MCU와 2GB DDR3를 갖추었으며, 임베디드 리눅스 OS를 설치하여 제스처 인식 알고리즘의 원활한 수행이 가능하다. 또한 차량 내부 네트워크를 위한 CAN 인터페이스가 있기 때문에 차량용 헤드유닛 플랫폼으로 적합하다.

본 논문의 제스처 인식 실험을 위해 총 6개의 제스처를 미리 정의했다. 정의한 제스처는 다음 <표 2>에 나타났다. 차량 내부에서 운전자가 쉽게 취할 수 있는 제스처 동작으로 구분지었다. <표 2>에 나타난 제스처 정보를 제외한 다른 동작은 무의미한 동작으로 판별하도록 했다. 모든 제스처를 실행하기 직전의 기본 상태는 착용한 팔의 팔꿈치를 굽혀 90도로 유지한 상태에서 시작했다.

2. 실험 데이터

웨어러블 디바이스 플랫폼을 이용한 차량 내 사용자 제스처 인식 성능에 대한 평가는 <표 2>에서 정의한 제스처 구분을 기반으로 진행했다. 본격적인 실험 진행에 앞서 특정 움직임 발생시 웨어러블 디바이스에서 발생하는 센서 값의 변동 예시를 <그림

표 2. 제스처 구분

제스처	동작	그림
① 왼쪽 비틀기	장치를 착용한 손목을 왼쪽으로 비틀기	
② 오른쪽 비틀기	장치를 착용한 손목을 오른쪽으로 비틀기	
③ 위로 굽히기	장치를 착용한 팔의 팔꿈치를 접어 위로 올리기	
④ 아래로 굽히기	장치를 착용한 팔의 팔꿈치를 펴서 아래로 내리기	
⑤ 시계방향 회전	장치를 착용한 팔의 팔꿈치를 축으로 시계방향으로 원을 그리며 회전	
⑥ 반시계방향 회전	장치를 착용한 팔의 팔꿈치를 축으로 반시계방향으로 원을 그리며 회전	

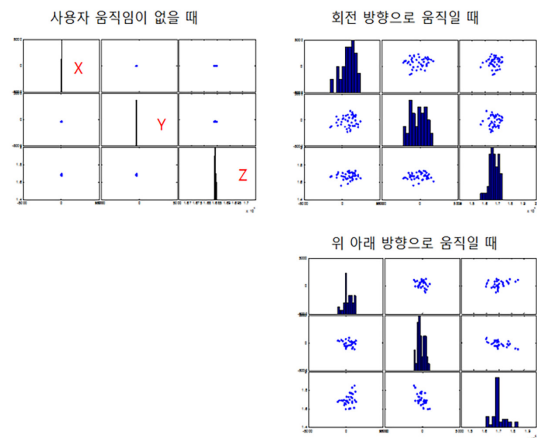


그림 9. 사용자 움직임에 따른 센서 값의 변화

9)에 나타났다. <그림 9>에서 볼 수 있듯이, 자이로 센서를 이용한 3축(x축, y축, z축) 대한 사용자의 제스처에 따른 값의 변화는 사용자의 움직임에 따라 다르게 나타남을 확인할 수 있다.

사용자의 움직임에 따라 변화되는 센서 값의 변동을 기반으로, 미리 정의된 센서 값 간의 상호상관 정보의 예를 다음 <표 3>에 나타냈다. <표 3>의 내용은 헤드유닛 플랫폼에서 웨어러블 디바이스 플랫폼의 센서 값을 수집하여 계산됐다. <표 3(a)>는 <표 2>에 나타난 제스처 구분의 '① 왼쪽 비틀기' 제스처 시의 상호 상관관계 정보이고, (b)는 '② 오른쪽 비틀기' 제스처의 정보이다. <표 3>의 1행과 1열은 순서대로 가속도 센서 X, Y, Z축, 자이로 센서 X, Y, Z축을 나타낸다. 표 안에 채워진 숫자는 각 센서간 상호 상관관계 값을 나타낸다. 예를 들어, 2행 3열의 값인 0.85는 가속도 센서 X축과 가속도 센서 Y축 변화

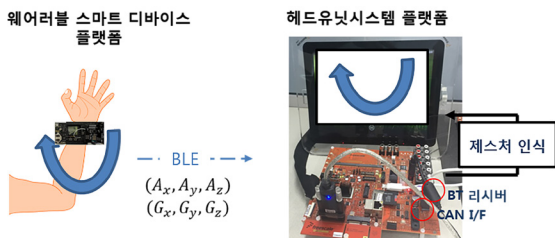


그림 8. 실험 환경 구성

값의 상호 상관관계가 0.85임을 나타낸다. <표 3(a)>, <표 3(b)>의 나머지 내용도 동일한 방식으로 작성되었다.

표로 정리된 상호 상관관계 정보에서 주목해야 할 부분은 표 3(a)의 3행 4열과 같이 1에 가까운 큰 값이다. 이는 ①번 동작시 가속도 센서 Y축과 Z축이 서로 큰 상관관계가 있다는 것을 의미한다. 이러한 큰 상호 상관관계 값의 위치를 기록하여 제스처 인식과 분류를 위한 DB로 사용 한다. 동일한 방법으로 표 3(b)의 가속도 센서 X축과 자이로 센서 Y간의 비교적 큰 상호 상관관계 값도 관찰된다. 이렇게 관찰되는 상호 상관관계 값을 기준으로, 본 논문의 알고리즘과 같이 사용자의 움직임 패턴을 파악하고 상호 상관관계 매칭으로 제스처를 인식한다.

표 3. 실험에 사용된 미리 정의된 상호 상관관계 값.

왼쪽 하단 내용은 오른쪽 상단의 내용과 대칭되므로 생략. A_x, A_y, A_z : 가속도 센서 X, Y, Z축, G_x, G_y, G_z : 자이로스코프 센서 X, Y, Z축

(a) ①번 동작시 상호 상관관계 정보

	A_x	A_y	A_z	G_x	G_y	G_z
A_x	1	0.85	0.74	0.37	-0.03	0.38
A_y		1	0.94	0.12	0.09	0.23
A_z			1	0.07	0.16	0.10
G_x				1	-0.74	0.44
G_y					1	-0.29
G_z						1

(b) ②번 동작시 상호 상관관계 정보

	A_x	A_y	A_z	G_x	G_y	G_z
A_x	1	0.78	-0.56	-0.22	0.8	-0.6
A_y		1	-0.70	-0.37	0.78	-0.49
A_z			1	-0.03	-0.61	0.31
G_x				1	-0.36	0.27
G_y					1	-0.47
G_z						1

3. 실험 결과

사용자의 움직임에 따라 웨어러블 디바이스에서 발생시키는 각 센서 축의 변화 값을 정규화하여 <그림 10>에 나타냈다. <그림 10>에 나타난 센서 값은 <표 2>의 ①-④의 제스처를 연속적으로 6번씩 수행한 시간에 따른 변화 값이다. 가장 위쪽의 그래프는 가속도 센서 X축의 시간별 변화를 나타내고, 그 다음 그래프는 가속도 센서 Y축의 시간별 변화를 나타낸다. 이와 같은 식으로 마지막에는 자이로 센서 Z의 값의 변화를 나타낸다. 총 6개의 그래프의 최초 1/4 부분의 그래프는 ① 동작을 나타내고 그 다음 1/4은 ② 동작을 나타내며, 순서대로 ③, ④ 동작의 변화를 나타낸다.

센서 값은 내부의 오류에 의해 무작위 잡음이 추가되기 때문

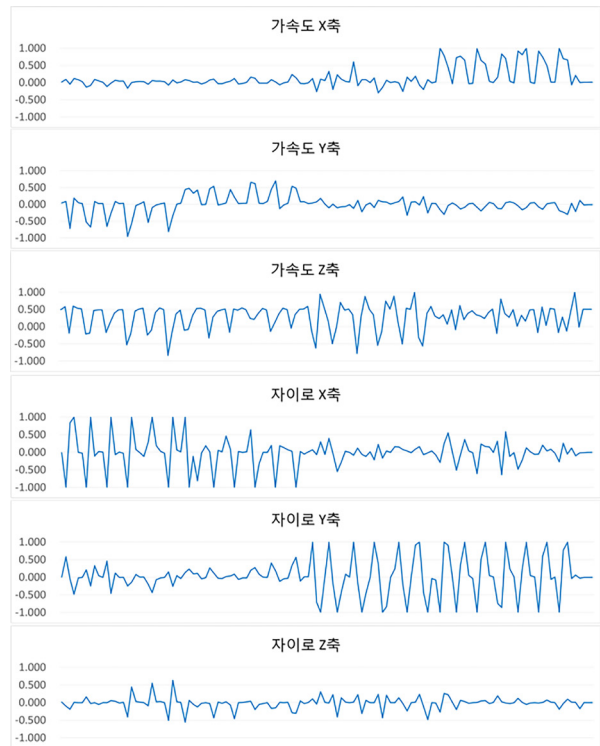


그림 10. 동작시 발생하는 센서 변화 값(정규화)

에 움직임이 없는 상황에서도 일정한 값을 정확히 유지하지는 않는다. 따라서 <그림 10>의 움직임 패턴에도 잡음이 추가된 값이 계속적으로 관찰된다. 이 상황에 본 논문에서 제시한 제스처 인식 알고리즘을 적용하면 모든 센서에 동시간에 발생하는 잡음은 무시되며 각 센서 축 간의 변화 관계로 움직임 파악이 가능해진다. <그림 10>에서 볼 수 있듯이, 연속된 움직임에서는 비슷한 양상의 센서 변화 값이 관찰된다. 특정 제스처 동작에 특정 축이 특정 수준 이상의 상호 상관관계에 놓여있음이 관찰된다. 즉, 잔 진동과 움직임이 발생하는 차량 내에서의 환경에서도 보다 높은 확률로 제스처 인식이 가능하다.

<그림 11>에 제안된 알고리즘이 적용된 제스처 인식 결과를 나타냈다. 인식률의 비교를 위하여 피크 값 구분으로 제스처를 구분 방법을 함께 나타냈다. 그래프의 x축은 <표 2>에 나타낸 6가지 제스처 종류를 나타내고 y축은 각 제스처의 인식률을 나타낸다. 제스처 별로 왼쪽의 막대 수치는 제안된 알고리즘이 적용된 경우를 나타내고 오른쪽 막대 수치는 비교를 위한 방법의 인식률을 나타낸다. 인식률 비교를 위하여 각 제스처 별로 30번씩 반복 수행했다.

<그림 11>에 비교를 위해 나타난 피크값 구분은 각 제스처 별로 특정 값 이상으로 큰 값이 나타나는 센서 축을 구분하여 해당 제스처로 매칭하는 방법이다. 이 방법은 특정 축에서 발생하는 피크 값을 기준으로 하기 때문에 방향성이 있는 동작의 구분

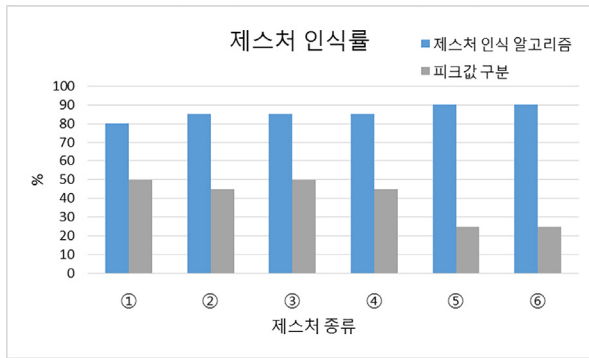


그림 11. 제스처 인식률

이 어렵다. 즉, 제스처 ⑤, ⑥의 구분이 어렵다. 또한 잡음이 많은 차량에서의 인식률은 보다 더 떨어진다. <그림 10>에서 볼 수 있듯이 제안된 알고리즘이 피크 값 기준 방법 보다 높은 제스처 인식률을 보이고 있음을 알 수 있다. 또한, 피크 값 구분 방법에서 판별이 어려웠던 제스처 ⑤, ⑥의 인식률이 높다. 이는 센서 값 간의 상호 상관관계를 파악하는 방법이 방향성을 갖는 동작에서도 인식률이 높음을 알 수 있다.

V. 결론 및 향후 연구

본 논문에서는 제스처 인식을 위한 밴드형 웨어러블 디바이스 플랫폼의 구현에 대해 설명했다. 웨어러블 디바이스 플랫폼의 구현과 제스처 인식 알고리즘에 대해 차례대로 설명했다. 웨어러블 디바이스 HW 플랫폼은 저전력 동작을 위해 설계됐으며, 움직임 센서 데이터의 수집과 헤드유닛 시스템으로의 데이터 전달을 주 기능으로 담았다. 제스처 인식 알고리즘은 수신된 센서 데이터를 상호 상관관계 계산을 이용해 가공하여 미리 정의된 DB와 매칭하도록 구성했다. 마지막으로 웨어러블 디바이스와 제스처 인식 알고리즘은 잡음과 방향성을 갖는 동작 인식에 강함을 실험을 통해 확인했다.

웨어러블 디바이스의 목적은 사용자에게 편의성을 제공하는 것에 있다. 웨어러블 디바이스를 커넥티드 카와 융합하여 생각해볼 때 궁극적으로 차량 내부의 조작 버튼을 제스처로 대체하는 것이 큰 의미를 가질 수 있다. 따라서 본 논문의 웨어러블 디바이스 플랫폼과 제스처 인식 알고리즘에 차량의 상세 정보를 함께 참조하는 차량 전용 제스처 인식 방법 연구를 진행할 계획이다. 제스처 인식에 필요한 주행정보, 속도, 급정거 이벤트 등의 수집이 가능한 차량정보수집 데이터와 웨어러블 디바이스의 움직임 정보를 함께 융합하여 보다 정확한 움직임을 인식하도록 알고리즘에 적용 할 것이다. 또한 이를 위하여 웨어러블 디

바이스에서 보다 많은 정보 처리를 해야하기 때문에 저전력으로 동작 가능하도록 전력 최적화 방향으로 연구를 지속할 예정이다.

Acknowledgment

본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업핵심기술개발사업의 일환으로 수행하였음. [10051106, 스마트키 기능을 포함하는 밴드형 웨어러블 디바이스 및 핵심기술 개발]

참고 문헌

- [1] 한상철, “웨어러블, 플랫폼화에 집중하자,” KEIT PD 이슈 리포트, JUNE 2015, Vol.15-6, pp.7-9
- [2] Ruize Xu, Shengi Zhou, and Wen J. Li, “MEMS Accelerometer Based Nonspecific-iUser Hand Gesture Recognition,” IEEE SENSORS Journal, Vol. 12, No. 5, MAY 2012, pp.1166-1173
- [3] Jaun Wu, Guifang Qiao, Jun Zhang, Ying Zhang, and Guangming Song, “Hand Motion-Based Remote Control Interface with Vibrotactile Feedback for Home Robots,” INTECH International Journal of Advanced Robotic Systems Vol.10, 270:2015, pp.1-10
- [4] Xuebing Yuan, Shuai Yu, Shengzhi Zhang, Chaojun Liu, and Sheng Liu, “Modeling and Analysis of Wearable Low-cost MEMS Inertial Measurement Units,” 15th International Conference on Electronic Packaging Technology, 2014, pp.542-546
- [5] 송재훈, 한정수, 오영준, 이희영, 변중남, “ARMA 모델을 이용한 근전도 신호의 특징추출,” 한국HCI학회 학술대회 2005, pp.30-35
- [6] Artem Demontyev and Joseph A. Paradiso, “Wrist-Flex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors,” ACM Article, Bibliometrics Data Bibliometrics, 2014, pp.1-6
- [7] X. Chen, X. Zhang et al., “Hand Gesture Recognition Research Based on Surface Sensors and 2D-accelerometers,” In Proc. of the 11th Int. Sympos. Wearable

Computers, 2007, pp. 11-16.

- [8] Fan Wei, Chen Xiang, Wang Wen-hui, Zhang Xu, Yang Ji-hai, "A Method of Hand Gesture Recognition based on Multiple Sensors," 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE), 2010, pp.1-4.
- [9] Ariful Islam Bhuyan, Tuton Chandra Mallick, "Gyro-Accelerometer based control of a robotic Arm using AVR Microcontroller," The 9th International Forum on Strategic Technology (IFOST), October 21-23, 2014, pp.409-413
- [10] Renqiang Xie, Xia Sun, Xiang Xia, and Juncheng Cao, "Similarity Matching-Based Extensible Hand Gesture Recognition," IEEE SENSORS JOURNAL, VOL. 15, NO. 6, JUNE 2015, pp.3475-3483

약 력



박인혜

2008년 광운대학교 공학사
 2010년 광운대학교 공학석사
 2015년 광운대학교 공학박사
 2012년~현재 전자부품연구원 임베디드, SW센터
 선임연구원
 관심분야: 임베디드시스템 SW, 차량용 임베디드
 시스템, 웨어러블 디바이스



이상엽

2003년 연세대학교 공학사
 2005년 연세대학교 공학석사
 2015년~현재 고려대학교 컴퓨터정보학 박사과정
 2005년~2009년 삼성전기 중앙연구소 선임연구원
 2009년~현재 전자부품연구원 임베디드, SW센터
 선임연구원
 관심분야: In-vehicle 네트워크 시스템, 차량용
 임베디드 시스템



고재진

1997년 광운대학교 공학사
 2000년 광운대학교 공학석사
 2013년 광운대학교 공학박사
 2000년~현재 전자부품연구원 임베디드, SW센터
 센터장
 관심분야: 웨어러블 디바이스, 임베디드 시스템