DBpia

논문 17-42-11-12 The Journal of Korean Institute of Communications and Information Sciences '17-11 Vol.42 No.11
https://doi.org/10.7840/kics.2017.42.11.2139

MEMS 센서를 이용한 상관 기반의 손 동작 인식 방법

박 인 혜◆, 이 상 엽°, 고 재 진*

Hand Gesture Recognition With Correlation Using MEMS Sensor

Inhye Park◆, Sang-yub Lee°, Jae-jin Ko*

요 약

본 논문은 밴드형 웨어러블 장치의 MEMS(micro-electromechanical system) 센서 정보를 이용한 상관 기반의 손 동작 인식 방법에 대해 서술한다. 제안한 시스템은 MEMS로부터 획득된 가속도와 자이로 센서 정보를 가공하여 데이터 베이스에 저장된 정보와의 해밍 거리(Hamming distance)를 계산한다. 계산 결과로 가장 작은 해밍 거리 값이 도출된 데이터 베이스 동작을 착용자의 움직임으로 판별한다. 비교를 위해 상관(correlation) 계산법을 기반으로 한 슬라이딩 상관방법(sliding correlation)과 상호 상관(inter-correlation) 방법을 사용한다. 여러 축이 동시에 변화되는 추세를 한 번에 비교하는 슬라이딩 상관은 비교적 단순한 동작의 매칭에 사용한다. 반면에 상호 상관은 동시에 다수 축에서 유효한 데이터 값이 발생하는 동작의 구분에 사용한다. 본 논문의 마지막에는 제안된 방법의 성능 평가를 위해 제작된 웨어러블 장치를 이용해 실착용 실험을 했다. 실험의 결과를 통해 제안된 동작 인식 방법이 기존 연구에 비해 훨씬 많은 동작을 구분함에도 불구하고 평균 97.1%로 높은 인식률을 가짐을 확인했다.

Key Words : Hand gesture recognition, MEMS sensor, acceleration sensor, correlation, wearable device

ABSTRACT

This paper presents a hand gesture recognition method using MEMS(Micro-electromechanical system) sensor for wearable devices. Proposed method computes the Hamming distance between the acquired data and stored data contained in database. The one gesture with the smallest distance is recognized the performed gesture. The system adjusted two correlation-based schemes that are sliding correlation and inter-correlation The sliding correlation comparing formulation trends is suitable for simple action matching. And the other is used to classify the more diverse gestures which have complex features. In the Section 4 on this paper, we described the experimental environments and results for evaluating the performance of proposed method. Especially, we used the test platform that manufactured as similar with wearable devices. From the experimental results, proposed method which include two matching scheme is evaluated by the notable recognition rate, 97%.

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

◆ First Author : Korea Electronics Technology Institute, ine.park@keti.re.kr, 정회원
° Corresponding Author : Korea Electronics Technology Institute, syublee@keti.re.kr, 정회원
* Korea Electronics Technology Institute, jaejini@keti.re.kr, 정회원
논문번호 : KICS2017-08-005, Received August 1, 2017; Revised September 29, 2017; Accepted October 12, 2017

www.dbpia.co.kr

Ⅰ. 서 론

스마트 기기가 지원해주는 많은 서비스 중에 하나인 HCI(human-computer interaction)[1]는 사람들에게 편리한 서비스 제공을 목적으로 한다. 가장 대표적인 것으로 웨어러블 글래스[2], 리모콘[3] 등이 있다. HCI 응용에서는 특히 사람의 손 동작을 인식하는 많은 연구가 진행돼왔다. 주로 이미지 인식과 MEMS(Micro-electromechanical system) 센서 기반 연구로 나뉜다.

이미지 인식을 이용한 동작 인식 응용에는 X-box[3], BMW 7시리즈의 에어 제스처[4]와 같이 널리 알려진 서비스가 많다. 이는 이미지 처리 분야와 같이 계산에 필요한 자원이 충분하고 빠른 시스템에서의 3D 인식을 위한 비전 인식 방법을 사용한다.[5,6] 대용량의 데이터를 빠르게 처리할 수 있으며 비교적 정교한 동작의 높은 인식률을 보인다. 하지만 반드시 카메라가 설치된 곳에서 움직여야 한다는 공간의 제약이 있고 빛 반사, 어둠의 원인으로 오인식의 가능성이 있다.[4-6]

MEMS 센서를 이용한 동작 인식은 비교적 인식률이 떨어진다는 단점이 있다. 하지만 동작 인식을 위해 필요한 장치의 가격이 저렴하고 전력 등의 자원 소모가 적으며 실행 공간의 제약이 없다는 장점이 있다.[7-9] 이를 이용한 대표적인 서비스는 웨어러블 장치의 만보계 기능, 운동 기능 등이 있다. 또한 웨어러블 장치를 이용해 HCI 기능을 제공하는 등 착용자에게 편의 기능을 제공할 수도 있다.

잡음이 많은 MEMS를 이용한 패턴 인식 분야에 정확도를 높이기 위한 많은 연구가 있었다. 확률 통계적인 수치를 이용해 매칭하는 방법[10], hidden Markov model(HMM)[11]을 이용한 방법이 연구돼왔다. 특히 매우 제한적인 하드웨어 자원을 이용해 빠른 시간 내에 동작 인식 결과를 내는 연구가 이뤄졌다.

가장 대표적인 연구로, 움직임 패턴의 시작 점을 찾아 매칭하는 dynamic time warping(DTW)[9] 방법이 있다. 저자는 3축 가속도 값을 이용해서 동작을 인식했으며, DTW와 affinity propagation algorithms 방법을 이용해 효과적인 학습 동작 인식 알고리즘을 제안했다. 해당 방법은 학습 동작 인식방법으로 실험을 진행할수록 보다 높은 인식률을 나타냈다. 그러나 제안한 동작 인식 개수가 한정적(8개)이고 시스템의 크기가 비교적 커서 임베디드 상황에서의 구현이 쉽지 않다는 단점이 있다.

R. Xu는 그의 논문[12]에서 Hopfield, 기반의 동작 인식 시스템을 제안했다. 속도의 증가가 특정 값 이상이 되면 움직임 발생으로 간주했다. 각 축별로 발생되는 유효한 크기의 센서 값의 부호를 각 축 별로 파악했다. 각 축별로 파악된 부호의 순서에 따라 동작을 매칭 했다. 해당 방법은 DTW와 마찬가지로 인식 가능한 동작 수가 비교적 적다(7개). 또한 센서 값 부호로 동작 인식을 진행하므로 비교적 미세한 움직임에서의 오인식 가능성이 있다.

다른 연구에서는 Smart ring[13] 제품에 직접 장치를 구현한 경우도 있다. 기본 동작 6종을 선정하여 별도의 학습 없이 기본 동작의 연속된 동작으로 추가 동작을 수행 가능하도록 구현했다. 해당 논문은 센서 값의 변화량을 측정하여 평균 변화량 값이 임계값 이상이 되면 시스템을 수행토록 했으며 Johnson code를 사용해 조합된 동작을 기본 동작으로 분리해 매칭 했다. 동작 인식을 한 축의 변화량으로 판단하기 때문에 한 번에 단일 축의 큰 움직이 포착된 제스처만 해석이 가능하다. 따라서 복합적인 움직임에 대한 동작 인식은 어렵다.

본 논문에서는 손목에 착용하는 MEMS 센서를 이용해 상관 기반의 동작 인식 방법을 제안한다. 또한 알고리즘의 성능 검증을 위한 실험에 대해 설명하고 그 결과에 대해 서술한다. 본 논문에서 제안한 슬라이딩 상관(sliding correlation)은 입력된 움직임 정보와 데이터 베이스 정보가 가진 각 축의 추세를 비교했다. 또 다른 제안 방법인 상호 상관(inter-correlation)은 각 축에 대한 서로의 관계를 비교하는 방법으로, 유사한 모양과 크기를 갖지만 그 방향이 다른 동작들을 구분해 내는데 사용했다. 제안한 방법으로 모양, 방향 등이 비슷한 유사도 높은 제스처를 구분해 낼 수 있었다. 또한 정교한 구분이 가능함에 따라 기존 연구들에 비해 구분 가능 제스처 개수를 높였다.

본 논문의 구성은 다음과 같다. 바로 다음에 이어질 2장에서 구현된 시스템 전체 구조와 센서 및 동작 특징에 대해 설명한다. 3장에서는 제안하는 알고리즘 방법에 대해 상세하고 설명한다. 4장에서 제안된 알고리즘의 실험을 위한 환경설정 정보와 실험 결과를 나타낸다. 마지막으로 5장에서 논문의 결론과 향후 연구 방향에 대해 언급한다.

Ⅱ. 시스템 구성

본 장에서는 제안한 방법을 적용하기 위한 시스템 구성에 대해 설명한다. 사용한 센서 데이터의 특성에 대해서 설명하고 동작 인식의 기준이 되는 동작의 개수와 구분에 대해 설명한다.

www.dbpia.co.kr

2.1 센서 데이터 설명

제안하는 동작 인식 방법은 동작 인식에 MEMS 센서를 사용한다. 제안한 시스템은 가속도 3축, 자이로 3축의 6측 센서를 사용한다. 가속도 센서는 3차원의 각 축에서 발생하는 속도 정보를 생성하고 자이로 정보는 동일한 3축에서 발생하는 각속도 정보를 나타낸다. 이 센서는 주로 항법, 네비게이션, 휴대전화에 장착되어 위치와 움직임 파악에 사용된다.

동작 인식 방법 평가를 위해서 웨어러블 장치와 유사한 환경의 실험 플랫폼을 제작했다. 또한 제작한 플랫폼에 제안한 동작 인식 프로그램을 구현했다. 가속도와 각속도 데이터를 20Hz로 표본을 추출했고 I2C를 통해 2 바이트 크기의 변수에 정수 형태로 저장했다. 따라서 수집된 정보는 2 바이트 로 표현될 수 있는 최솟값 및 최댓값인 -32,768에서 32,767까지의 범위를 갖도록 했다. 움직임이 포착되면, 센서 데이터는 포착된 시점부터 1초 동안 발생한 표본을 하나의 집합으로 엮어 동작 인식을 진행했다.

2.2 동작 소개 및 분류

동작 인식의 정확성 확대를 위해 모든 동작은 1초 동안 수행 돼야하고 수평인 정지 상태에서 시작돼야 한다. 움직임 여부를 포착하기 위해 센서 값이 특정 임계값을 넘은 시점을 기준으로 연속 표본을 수집한다. 수집된 표본은 전처리 과정을 거친 뒤, 저장된 데이터 베이스와 일련의 방법으로 유사도를 검사한다.

본 논문에서는 인식할 동작의 종류를 36개로 선정했다. 36개의 동작을 다시 3개의 그룹으로 나눴다. 6개의 단순 동작으로 이뤄진 SIMPLE, 연속된 SIMPLE 동작의 조합으로 볼 수 있는 24개의 COMBINED, 다축에서 동시에 움직임이 발생하는 6개 CIRCLE이다. 이들 동작과 구분이 그림 1에 나타나있다. COMBINED 그룹의 동작은 SIMPLE의 연속 동작으로 오인될 수 있다. 따라서 연속적으로 동작을 수행할 때는 하나의 동작 다음에 1초간의 유휴 기간을 둬야한다.

Ⅲ. 밴드형 웨어러블 장치를 이용한 동작 인식

이 장에서는 동작 인식 방법을 상세 설명한다. 손목에 착용된 웨어러블 장치로부터 얻은 움직임 정보를 이용해 전처리, 정보 해석, 데이터 베이스 매칭의 순서로 동작 인식을 진행한다. 각 순서에 대한 설명과 그룹별로 진행되는 동작 인식 방법에 대해서 순서대로 설명한다.

3.1 데이터 수집과 전처리

안정적인 동작 인식 방법 실행을 위해 발생되는 데이터 집합 지정을 선행한다. 웨어러블 장치에 부착된 MEMS 센서는 끊임 없이 추출 값을 출력한다(본 논문에 사용된 센서는 초당 20Hz로 표본추출 주기를 설정). 그 중 움직임의 시작점을 찾아 그 시점부터 1초간의 데이터를 묶어 동작 인식을 진행해야 한다. 센서 값을 계속적으로 관찰하다가 특정 임계값을 넘으면 데이터 수집과 전처리 과정을 시작한다.

전처리 과정의 첫 번째 단계는 센서 데이터의 잡음과 오류 저감을 위한 정규화 및 슬라이딩 창 평균(SWA: sliding window average)의 적용이다. 정규화는 2 바이트로 표현 가능한 최댓값인 32,767으로 진행했다. 특정 창 만큼의 숫자를 시프트하며 평균을 구하는 SWA는 다음 식 1로 표현된다. $y(i)$는 SWA로 전처리된 데이터 세트의 i번째 원소를 의미한다.

$$y(i) = \frac{1}{f_s} \sum_{k=0}^{f_s - 1} x_{i+k}, \qquad i < N_s \tag{1}$$

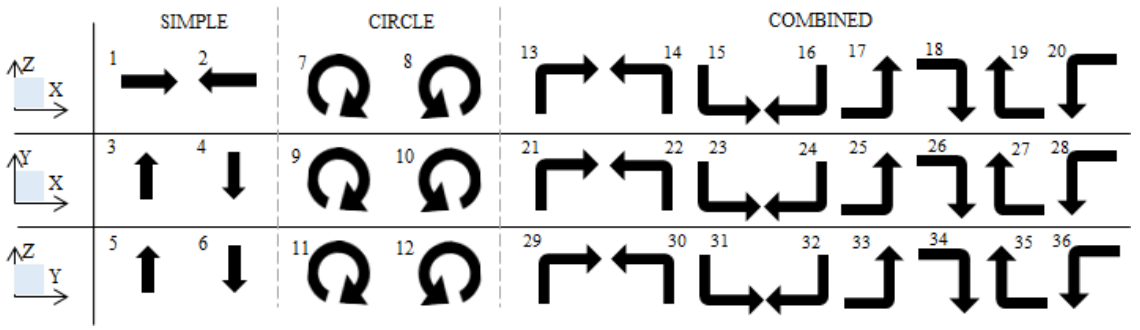

그림 1. 손 동작 인식을 위해 분류한 동작 종류
Fig. 1. Gestures to recognize hand gestures in our system

www.dbpia.co.kr

N은 데이터 집합에 포함된 표본의 숫자, f_s는 얼마만큼 SWA를 병합할지에 대한 창 변수, N_s는 SWA를 수행하고 난 뒤의 데이터 집합 표본 수를 나타낸다. 따라서, $N_s = N - f_s$로 나타낼 수 있다. 본 논문에서는 f_s를 3으로 설정했다.

특정 임계값이 넘는 시점을 시작으로 1초간 동작 인식을 위한 데이터 표본을 수집한다. 움직임 정보는 직전의 위치에 따라 변하는 상대적인 정보이고, 같은 사람이 동일한 동작을 수행한다고 해도 명확히 동일한 센서 값이 발생되지는 않는다. 따라서 정확도를 높이기 위한 계산의 일환으로 잉여 표본을 함께 수집한다. 임계점을 넘는 순간과 그 직전에 발생된 일부 표본을 함께 데이터 집합으로 묶는다. 동작 인식을 수행하지 않더라도, 발생되는 추출 값을 지속적으로 임시 버퍼에 저장해 관찰 한다.

다음 그림 2에 임시 버퍼 저장에 대한 설명을 그림으로 나타냈다. 그림 2는 발생되는 센서 값을 시간에 따라 나타낸 그래프로, y축은 센서 값을 의미한다. 센서 값이 임계값을 넘는 p 지점에서부터 그 직전에 수집된 a개 만큼의 표본을 데이터 집합으로 포함한다. 즉, $p-a$ 지점부터 1초 동안 발생한 표본을 동작 인식 집합으로 사용한다. 데이터 집합이 선정되면, 그 다음엔 데이터 집합 내 최대 차이를 계산한다. 즉, 최댓값(M)과 최솟값(m)의 차를 계산해서 미리 선정된 값 이상일 경우에만 동작 인식 과정을 진행한다. 이와 같은 과정은 미세한 움직임이나 순간적 잡음으로 유발되는 불필요한 동작 인식 모듈 수행을 막을 수 있다.

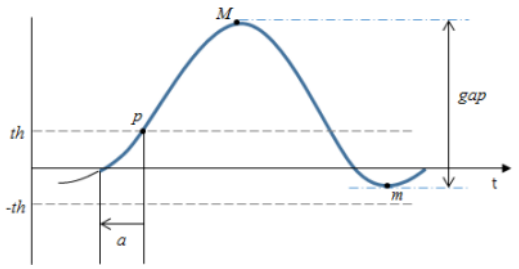

그림 2. 표본추출과 gap 계산 방법
Fig. 2. Sampling and gap schemes

3.2 동작 분류

전처리 과정을 마친 뒤에, 측정된 차이 값에 따라 동작 그룹 별 분류를 진행한다. 차이 값 분류는 가속도 3축 정보를 이용한다. SIMPLE의 경우에는 단일 축을 향한 동작이기 때문에, 단일 축에서만 차이가 클 경우는 모두 SIMPLE 그룹으로 분류한다. 예를 들어, X축에만 유효 값이 발견될 경우에는 *오른쪽* 혹은 *왼쪽*의 데이터 베이스와 비교하여 동작 동작을 인식한다. 그룹 분류는 기저장된 전체 동작과 일일이 비교하는 것보다 자원을 훨씬 적게 소모한다.

두 축 이상에서 동시에 유효한 값이 발견될 때는 추가적인 계산을 한다. COMBINE과 CIRCLE 그룹을 구분해야하기 때문이다. 그림 3에 나타난 것과 같이 COMBINE과 CIRCLE 그룹 동작들은 유효한 값이 발생하는 축이 동일하며, 동작의 방향도 비슷하기 때문에 구분이 쉽지 않다. 그림 3(a)의 왼쪽 그림은 CIRCLE의 *시계방향* 동작이고, 오른쪽 그림은 COMBINED의 *위-오른쪽* 동작이다. 중력에 반하는 방향으로 움직이고 그 직후 혹은 동시에 오른쪽 방향으로 이동한다는 점이 매우 유사하다. 그림 3(b)에는 두 동작을 수행할 때 발생하는 가속도와 자이로 센서 값을 그래프로 나타냈다. 그림 3(b)에서 볼 수 있듯이, 두 동작은 유효한 움직임 축의 종류(가속도 x축과 z축)와 센서 값 변화 모양이 매우 유사하다.

두 축에서 동시에 유효한 값 발생이 관찰되면 절정 값의 위치를 기준으로 각 축마다 Mm 블록을 선정한다. 절정 값의 위치는 그림 3(b)에서 찾을 수 있다. 센서가 최댓값과 최솟값으로 변화는 구간을 Mm 블록으로 지정하고, 각 축마다의 Mm 블록 위치의 겹침 정도를 파악한다. 이를 통해 두 축 이상이 변화가 동시에 진행되는지, 순차적으로 일어나는지를 판단한다. 임계값 이상으로 값이 발생한 축마다 Mm 블록을 계산하여 서로 겹치는 Mm 블록의 표본 수가 특정 개수 (f_o로 정의) 이상 이면 CIRCLE 그룹으로, 반대로

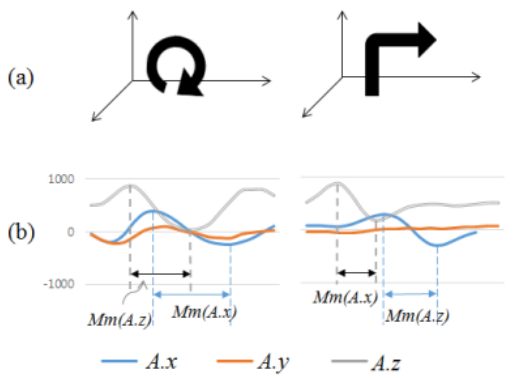

그림 3. 시계방향과 위-오른쪽 동작과 각 축 그래프
Fig. 3. Clock and up-right gestures and axes

www.dbpia.co.kr

f_o 개 미만이면 COMBINED 그룹으로 분류한다. 이러한 방법으로 입력된 동작의 그룹을 먼저 분류한 후에 동작 매칭 모듈을 수행한다.

3.3 SIMPLE & COMBINED

SIMPLE과 COMBINED 그룹의 동작 구분에는 슬라이딩 상관 방법을 사용한다. 상관을 사용하여 가속도 변화 추세를 비교한다. 즉, 동작의 가속도 변화 특징과 가장 유사한 데이터 베이스로 동작을 매칭을 한다. 상관은 추세 비교를 할 때 그 값이 -1과 1 사이의 실수로 나타나기에 다양한 값을 얻을 수 있다. 그러나 추세 특징이 나타나는 구간의 범위를 명확히 정하지 않으면 유효한 상관이 불가하다.

손목 착용 보드를 이용한 동작 수집은 데이터 집합을을 정확히 맞추기가 어렵다. 사용자가 매번 동일한 속도와 크기로 동작을 수행한다고 완전히 똑같은 센서 값의 집합을 수집하기란 거의 불가능하다. 이런 단점을 보완하기 위해 본 논문에서는 특정 지점을 시작 구간으로 선정하고 그 주변의 표본들을 함께 확인하는 슬라이딩 상관을 제안한다. 슬라이딩 상관은 f_c라는 변수만큼 한 칸씩 시프트되어 각 데이터 베이스와 수 회 상관관계 매칭하는 방법이다. 예를 들어 f_c가 3이고 매칭 대상이 되는 데이터 베이스(DB)가 3개라면, (3번 오른쪽 시프트 + 3번 왼쪽 시프트 + 1) * DB 3개 = 21번의 상관이 일어난다. 21개의 상관 결과 값 중 가장 높은 최댓값이 위치하는 데이터 베이스를 인식된 동작으로 선정한다.

설명된 슬라이딩 상관의 표현을 위해서 다음 몇 가지를 정의한다. Y는 전처리된 데이터들이 1번부터 N_s번까지 순서대로 배열된 집합이고, $Y = \{y(1), y(2), ..., y(N_s)\}$로 표현된다. Y'는 Y의 부분집합이다. $Y'_{a,b}$는 a번째 원소부터 시작해 b원소까지 나열된 부분집합이다. 예를 들어 $Y'_{1,3} = \{y(1), y(2), y(3)\}$. 앞서 설명됐듯이, 데이터 베이스에 저장된 데이터 집합과 입력된 데이터 집합은 f_c개의 표본씩 겹쳐 상관비교한다. f_c개의 겹침 비교를 진행한 슬라이딩 상관 결과의 집합을 SC라고 하면, SC는 다음 식 2와 같이 표시된다.

$$\begin{aligned} SC = [&\rho(Y'_{1,N_S-f_C}, DB'_{f_C,N_S}),\\ &\rho(Y'_{1,N_S-f_C+1}, DB'_{f_C+1,N_S}),\\ &\cdots,\\ &\rho(Y'_{1,N_S}, DB'_{1,N_S}),\\ &\cdots,\\ &\rho(Y'_{f_C,N_S}, DB'_{1-N_S-f_C})] \end{aligned} \tag{2}$$

DB'는 DB의 부분집합이고 $DB'_{a,b}$는 $Y'_{a,b}$와 같은 규칙을 따른다. 위 식에서 알 수 있듯이, DB와 Y를 각각 f_c가 될 때까지 원소를 한 칸씩 시프트되어 상호 비교함을 알 수 있다. 데이터의 상호 비교는 하나의 데이터 집합 당 {(f_c * 2 + 1) * 대상 DB} 만큼 이뤄진다. 가장 높은 상관관계입력 값이 수행된 동작으로 선정되는 DB 결정 방법은 식 3과 같다. 아래 식 3에서 k는 비교될 동작의 색인을 의미하며, 색인의 숫자는 그림 1에 명시된 번호를 따른다.

$$match = \underset{DB_k}{\arg\max}\, SC_k \tag{3}$$

3.4 CIRCLE

CIRCLE 그룹의 동작 인식에는 상호 상관 방법을 제안해 적용했다. 상호 상관은 수집된 데이터 집합 안에서 각 축 간 상호 증감 관계를 계산한다. 이 방법은 수행하는 모양은 유사하지만 방향이 다른 동작을 구분하기에 유리하다. 즉, 그림 1의 7번과 8번처럼 움직이는 동작의 크기, 수행의 빠르기, 제스처 모양 등은 유사하지만 그 방향이 반대인 경우를 구분하기에 유리하다. 각 축에서 시간의 흐름에 따라 발생하는 증감 추세가 다르기 때문에 축 간의 상관 값이 상이하다. ρ_{ij}를 동일 시간에 수집된 센서 i와 j 값의 상관 값이라 할 때($\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$), 수집된 센서 값들의 상호 상관 행렬인 A는 다음 식 4와 같이 표현된다.

$$A = \begin{bmatrix} 1 & \rho_{A_xA_y} \cdots & \rho_{A_xG_z} \\ \rho_{A_yA_x} & 1 & \cdots \rho_{A_yG_z} \\ \vdots & \vdots & \ddots \vdots \\ \rho_{G_zA_x} & \rho_{G_zA_y} \cdots & 1 \end{bmatrix} \tag{4}$$

식 4에서 A_x, A_y, A_z는 가속도의 x, y, z 축을,

www.dbpia.co.kr

G_x, G_y, G_z는 자이로 x, y, z축을 각각 의미한다. 행렬 A에서 자기 자신과의 상관 값은 1이기 때문에 $i=j$인 대각 요소들은 모두 1이다. 또한 $\rho_{ij}=\rho_{ji}$므로, 대각선을 기준으로한 대칭 행렬이다.

먼저 1초 동안 입력된 데이터 집합을 이용해 A를 계산한다. A 행렬 요소 해밍 거리 합이 최소가 되는 DB_k를 찾는다. 계산 편의를 위해 절대 값이 유효한 수치 이하일 경우는 해밍 거리 값을 삭제한다(본 논문에서는 0.5로 설정했다.). 0이 아닌 값 중 각 동작 비교에 유효한 원소 위치로 이뤄진 집합인 U_m을 정의한다. 즉, 유효한 절대 값 중 가장 큰 C_n개의 값을 선정하여 집합으로 재구성한다. U_m에 삽입될 원소는 다음의 식 5에 의해 A 행렬의 원소를 대상으로 결정한다.

$$U_m \ni \arg\max_{i,j}(|[\rho_{i,j}]|), \quad A = A - U_m,\ 0 \notin U_m \tag{5}$$

최초의 U_m은 공집합에서 시작된다. U_m은 시스템에서 미리 정의한 C_n값 만큼 반복되어 원소가 쌓인다. 데이터 베이스 속 모든 동작은 각자의 A 행렬을 통해 계산된 U_m 집합을 가지며, 수집된 데이터의 U_m을 비교해서 동작을 매칭한다. 즉, 특정 동작 U_m에서 동일한 위치의 요소의 거리가 가장 작은 것이 매칭된다. 예를 들어, *시계방향* 동작과 *반시계방향 동작*을 비교해 볼 수 있다. *시계방향* 동작의 유효한 상호 상관관계 값은 $U_7=(A_xA_y,\ A_yA_z,\ G_xG_y)$이며, 원소는 $A_xA_y=0.85$, $A_yA_z=0.94$, $G_xG_y=$ -0.74이다(단, $C_n=3$). 반면에 *시계방향* 동작은 $U_8=(A_xA_y,\ A_yA_z,\ A_xG_y)$, $A_xA_y=0.78$, $A_yA_z=-0.56$, $G_xG_y=0.8$의 값을 갖는다. 여기서 주목할 점은 A_xA_y, A_yA_z를 동일한 원소로 갖으며 각 동작의 A_yA_z는 서로 부호가 다른 값이라는 점, 두 동작이 각각 G_xG_y와 A_xG_y라는 다른 원소를 갖는다는 점이다. 이에 주목하여 입력 된 데이터 값을 DB_k와 비교하여 해밍거리가 최소인 동작을 도출한다. 도출된 동작은 착용자가 수행한 동작으로 매칭된다.

Ⅳ. 실험 및 결과

이 장에서는 제안한 방법의 성능을 알아보기 위하여 꾸며진 환경과 주어진 환경에서 수행된 실험의 결과를 설명한다.

4.1 환경 설정

제안한 동작 인식 알고리즘의 실험을 위해 제작한 하드웨어 플랫폼에 대해 설명한다. 실험을 위해 손목 착용이 가능한 크기로 임베디드 수준의 웨어러블 장치 플랫폼을 제작했다. Main computing unit(MCU)은 ARM Cortex-M3를 기반의 EFM32 모델을 장착했고, 무선통신 지원을 위해 Silicon Lab.의 Bluegiga BT121모델을 이용해 Bluetooth low energy(BLE)을 지원한다. MEMS 센서는 9축 정보를 발생시키는 InvenSense의 ICM-20948 모델을 장착했다. 특히 MEMS 센서는 9축 정보 중에 6축의 정보만을 16비트씩 추출하여 사용했다. 추출된 데이터는 A/D 컨버터를 통해 디지털 신호로 변환됐다. ICM-20948은 변환된 추출 신호를 I2C를 이용해 MCU로 전달한다. 전달된 추출 데이터를 이용해 본 문에 제안된 동작 인식 알고리즘을 수행했다. 개발보드 사진과 착용 모습을 그림 4에 나타냈다.

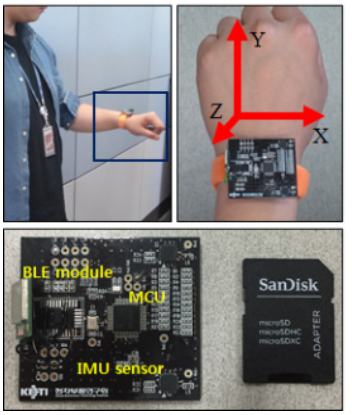

그림 4. 제작된 웨어러블 장치와 착용 모습
Fig. 4. Wearable device for testing

4.2 실험 결과

동작 인식률 검증을 위해 그림 1에 명시한 동작 36종을 반복 수행했다. 3명의 참가자가 각 동작 당 140회씩 반복하여 2주간 총 5,040번의 실험을 진행했다. 각 참가자는 자신의 왼쪽 손목에 장치를 장착했으며 장치의 상태를 수평으로 유지한 상태로 동작을 수행

www.dbpia.co.kr

표 1. 동작 그룹 인식 률
Table 1. Recognition rate of gesture groups

Input gestures	Recognized gestures: SIMPLE	CIRCLE	COMBINED	No. of test Sample
SIMPLE	99.7	0.3	0.0	840
CIRCLE	0.1	97.4	2.5	840
COMBINED	0.1	1.3	98.6	3,360

했다. 실험이 진행되는 동안에 움직임 임계값까지 도달하지 않아 동작 인식이 실행되지 않은 경우는 결과에 반영하지 않았다. 동작 인식 알고리즘의 성능을 파악하는 것이므로 일단 알고리즘이 실행되어야하기 때문이다.

표 1에 그룹별 동작 인식 정확도를 나타냈다. 수행된 움직임이 의도된 동작으로 매치됐을 경우와 의도된 동작은 아니지만 동일한 그룹의 동작으로 매칭된 경우 모두를 의미한다. 즉, 수행된 동작이 완벽 매치가 아닐지라도 적어도 동일한 그룹으로 분류 됐음을 의미한다. 표 1에 의하면 세 그룹 각각 99.7%, 97.4%, 98.6%로 그룹 내 매칭 정확도로 매우 높은 수준임을 알 수 있다.

표 2에는 본 논문에 제안된 알고리즘을 이용해 36종 동작의 상세 인식률을 나타냈다. 또한 각 동작을 그룹으로 구분하여 그룹별 평균 인식률을 나타냈다. 동작 1, 2의 인식률은 99.29%이고 3번 동작의 인식률은 98.57로 측정됐다. 또한 6개의 동작으로 이뤄진 SIMPLE 그룹의 평균 인식률은 98.57%로 측정됐다. 이와 동일한 방식으로 나머지 33개의 동작의 인식률을 나타냈으며, CIRCLE 그룹은 95.95, COMBINED 그룹은 96.7%로 측정됐다. 가장 높은 인식률은 동작 1,2이며, 가장 낮은 인식률은 CIRCLE 그룹 10번 동작의 91.43%이다. 가장 낮은 동작 인식률도 91%를 선회함을 확인했다.

표 3. 기존 연구와의 결과 비교
Table 3. Comparison with previous researches

Algorithm	No. of Gestures	Accuracy (%)	No. of test Samples
DTW [9]	8	98.6	540
SSTM [12]	7	95.6	628
SM [13]	12	97.2	1,134
Proposed Algorithm	36	97.11	5,040

표 3에 제안된 동작 인식 알고리즘과 이전 동작 인식 연구의 인식률을 비교했다. 각 수치는 이전 연구 논문에 나타난 정보로 작성됐다. 비교 대상은 이전 연구인 DTW[9], Sign Sequence and Template Matching (SSTM)[12], Gesture Decomposition and Similarity Matching (SM)[13]이다. 기술된 각 알고리즘은 수행된 하드웨어 플랫폼 성능, 동작 환경 및 방법, 수행 주체, 수행 제스처 모양과 종류 등이 다르다. 본 논문에서의 실험 결과와 직접 비교는 어려울 수 있다. 다만 참고를 위해 인식 정확도와 함께 실험 시도 횟수를 함께 언급했다. 표의 가장 왼쪽에는 알고리즘 구분을 나타냈고 다음 칸엔 해당 알고리즘에서 구분했던 동작의 개수, 그 다음엔 인식률, 그리고 가장 오른쪽에는 반복 실험 횟수를 나타냈다.

표 3에 따르면. 인식률이 DTW가 가장 높은 것으로 나타냈지만 제안한 알고리즘이 4배 이상 많은 동작을 구분 할 수 있다. 인식할 수 있는 동작이 많다는 것은 유사한 동작들을 보다 정교하게 구분해 낼 수 있기 때문에 보다 정확도가 높다고 볼 수 있다. 제안한

표 2 전체 동작 인식률
Table 2. Gesture recognition rate

Group	SIMPLE (98.69)						CIRCLE (95.95)					
No. of gesture	1	2	3	4	5	6	7	8	9	10	11	12
Accuracy	99.29	99.29	98.57	97.86	98.57	98.57	97.86	97.14	95.00	91.43	97.14	97.14
Group	COMBINED (96.7)											
No. of gesture	13	14	15	16	17	18	19	20	21	22	23	24
Accuracy	98.57	97.86	98.57	97.86	97.86	98.57	95.71	98.57	95.00	97.86	96.43	92.86
Group	COMBINED (continued)											
No. of gesture	25	26	27	28	29	30	31	32	33	34	35	36
Accuracy	97.86	98.57	97.14	97.86	96.43	96.43	92.14	98.57	95.00	96.43	96.43	92.14

www.dbpia.co.kr

알고리즘은 인식 가능 동작 개수가 훨씬 많음에도 인식률 차이는 1.5%의 매우 작은 수준임을 알 수 있다.

V. 결 론

본 논문은 손목에 착용하는 MEMS 센서를 이용한 동작 인식 알고리즘을 제안했다. 또한 알고리즘의 성능 검증을 위한 실험에 대해 설명하고 그 결과에 대해 서술했다. 웨어러블 장치 플랫폼을 착용 후 손 동작을 취해 그 동작을 인식하도록 했다. 수집된 센서 값은 정규화, SWA의 전처리 과정을 거치며 잡음이 제거됐다. 전처리 과정이 완료된 센서 값을 분류하여 특정 동작 그룹 별로 동작 인식 방법을 다르게 적용했다. 본 논문의 마지막에는 제작된 웨어러블 장치 플랫폼을 이용한 동작 인식 정확도 실험을 진행했다. 이에 높은 유사도를 지닌 제스처를 정확하게 구분하고 기존 연구보다 더 많은 개수의 제스처를 구분함을 확인했다. 총 36종의 동작을 140회씩 수행한 인식 결과로 평균 97.11%의 정확도를 도출했다. 제안된 알고리즘이 유효한 수준 이상의 높은 인식률을 가짐을 확인했다.

본 논문에서 제안 및 구현된 임베디드 소프트웨어는 HCI와 관련된 여러 응용에서 사용될 수 있다. 하나의 응용으로, 착용자의 주의 집중을 방해하지 않도록 차량과의 인터페이스 역할을 할 수 있다. 터치식으로 동작하는 기존의 인터페이스는 정확한 지점을 선정해야한다. 그러나 본 논문에서 제안한 시스템으로 지정된 동작을 특정 기능과 매칭하여 긴급하지 않은 명령 등을 실행시킬 수 있다. 이를 테면, 전화받기, 오디오 볼륨 향상 등의 기능이다.

본 논문의 저자는 향후 연구로 웨어러블 장치가 스스로 학습하는 최적화된 소프트웨어와 표본 특징을 추출하는 알고리즘을 설계 및 구현하고자 한다. 이를 통해 기존 36개로 정형화된 동작 뿐 아니라 사용자가 원하는 동작을 추가하여 보다 사용자 친화적인 서비스를 제공할 기반 기술을 개발 할 예정이다.

References

[1] E. Sato, T. Yamaguchi, and F. Harashima, "Natural interface using pointing behavior for human-robot gestural interaction," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1105-1112, Apr. 2007.

[2] M. Kim, J. Hong, M. Cho, and J. Choi, "Study on optimal information presentation area in smart glasses," in *Proc. HCI Korea 2017*, pp. 109-112, Feb. 2017.

[3] R. Francese, I. Passero, and F. Tortora, "Wiimote and Kinect: gestural user interfaces add a natural third dimension to HCI," in *Proc. Int. Working Conf. Advaned Visual Interfaces*, pp. 116-123, May 2012.

[4] S, Wu, T. Gable, K. May, Y. Choi, and B. Walker, "Comparison of surface gesture and air gestures for in-vehicle menu navigation," *Article of Archives of design research*, vol. 29, no. 4, Nov. 2016.

[5] S. Berman and H. Stern, "Sensors for gesture recognition systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 3, pp. 277-290, May 2012.

[6] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1685-1699, Sept. 2009.

[7] J. K. Oh, et al., "Inertial sensor based recognition of 3-D character gestures with an ensemble classifiers," in *Proc. IWFHR*, pp. 112-117, Oct. 2004.

[8] S. Zhou, Z. Dong, W. J. Li, and C. P. Kwong, "Hand-written character recognition using MEMS motion sensing technology," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, pp. 1418-1423, Jul. 2008.

[9] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6197-6205, Dec. 2011.

[10] D. H. Rubine, "The automatic recognition of gesture," Ph.D dissertation, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, Dec. 1991.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st Ed., New York: Springer, 2006.

[12] R. Xu, S. Zhou, and W. J. Li, "MEMS accelerometer based nonspecific user hand gesture recognition," *IEEE Sensors J.*, vol. 12,

www.dbpia.co.kr

no. 5, pp. 1166-1173, Nov. 2012
[13] R. Xie, X. Sun, X, Xia, and J. Cao, “Matching-based extensible hand gesture recognition,” *IEEE Sensors J.*, vol. 15, no. 6, pp. 3475-3483, Jun. 2015

박 인 혜 (Inhye Park)

2008년 : 광운대학교 공학사
2010년 : 광운대학교 공학석사
2015년 : 광운대학교 공학박사
2012년~현재 : 전자부품연구원 임베디드SW센터 전임연구원
<관심분야> 임베디드시스템 SW, 차량용 임베디드 시스템, 웨어러블 디바이스

이 상 엽 (Sang-yub Lee)

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

고 재 진 (Jae-jin Ko)

1997년 : 광운대학교 공학사
2005년 : 광운대학교 공학석사
2013년 : 광운대학교 공학박사
2000년~현재 : 전자부품연구원 임베디드SW센터 센터장
<관심분야> 웨어러블 디바이스, 임베디드 시스템

www.dbpia.co.kr