

논문 2011-06-10

# 실측 데이터 기반 모션센서 에뮬레이터의 개발

## (Development of Real-Data Motion Sensor Emulator)

이 민 석\*  
(Minsuk Lee)

Abstract : This paper describes the development of an open source motion sensor emulator. It helps developers to understand the motion sensor and its data better. Through this emulator, the realtime or stored motion sensor data can be applied to the applications that utilize motion sensors. The data of motion sensors which include accelerometer sensor, magnetic field sensor, gyro sensor, GPS, and so on, can be collected by using smart phones or motion sensors. We also describe a visualizer which shows various graphs, video and 3D animations based on the data sent by the emulator. It helps developers to understand motion sensors and how to use the sensors. The developed emulator is compatible with Android sensor simulator.

Keywords : Motion sensor, Emulator, Filter, Visualizer

### 1. 서 론

모션센서는 물리적인 움직임을 감지하기 위한 센서들을 통칭하는 것으로, 통상적으로 가속센서 (Accelerometer), 자이로센서 (Gyroscope Sensor), 자기장센서 (Magnetic Field Sensor), 근접센서 (Proximity Sensor) 등을 일컫는다. 또 움직임에 의한 위치의 변동을 측정할 수 있는 GPS도 모션센서의 일종으로 볼 수 있다. 비교적 최근까지 모션센서는 압전 소자, 기계적인 부품, 코일 등과 같은 개별 부품이 조립되어, 미세한 전압 변화 또는 자기장의 변화를 측정하도록 구현되었다. 그 결과 모션센서들은 상당히 크기가 크고, 부품의 가격도 비싼 상태였으며, 응용 범위도 비행기, 선박, 우주선의 항법 장치, 군수용 장비 등 제한적이었다. 하지만 1990년대 중, 후반부터 급속한 발전을 해온 MEMS (Micro-Electro-Mechanical Systems) 반도체 기술에 힘입어 [1], 최근에는 이러한 모션센서를 매우 작은 하나의 반도체 부품으로 구현하기에

이르렀으며 그 가격도 매우 저렴해졌다. 현재 모션센서는 게임기인 Wii를 시작으로 안드로이드폰과, 아이폰 등의 스마트폰에 경쟁적으로 적용되어 매우 중요한 입력 도구로써 널리 사용되고 있으며 [2], 2011년에는 전 세계 MEMS 기반 모션센서 시장 규모가 720억불에 이를 것으로 예측되고 있다 [3].

모션센서는 물리적인 움직임을 가속도, 각속도, 자기 방성, 위치 등을 수치화하는 장치이다. 이러한 수치의 의미를 해석하여 움직임(즉, 제스처)을 인식한 결과는 게임기에서 캐릭터의 움직임으로 표현되거나, 스마트폰과 같은 장치에서 특정 기능을 수행하도록 지시하는 사용자 인터페이스 입력으로 이용된다. 최근에 각광을 받고 있는 증강현실 응용의 경우, 스마트폰, 또는 HMD (Head Mounted Display)를 이용하여, 실제 카메라로 촬영한 영상 위에, 영상과 관련이 있는 부가 정보를 보여줌으로써 현실에 대한 이해를 늘린다. 이 증강현실 응용이 가능하려면, 영상이 출력되는 기기의 위치와 방향 정보가 정확하게 소프트웨어에 전달되어야 하는데, 이를 위해서도 모션센서가 사용된다. 그밖에도 모션센서의 값이 저렴해짐으로 인하여, 자동차의 여러 장치, 가전기기, 장난감 등 많은 임베디드 시스템에도 모션센서가 적극적으로 활용되고 있다.

스마트폰 응용 개발을 포함하여 모션센서를 활용하는 소프트웨어 개발이 점차 늘어나고 있는데 반하여, 많은 소프트웨어 개발자들의 모션센서 이해

\* 교신저자(Corresponding Author)

논문접수 : 2011. 01. 21., 수정일: 2011. 02. 10.,

채택확정 : 2011. 02. 14.

이민석 : 한성대학교 컴퓨터공학과

※ 본 논문은 2010년 추계학술대회 우수논문임

※ 본 연구는 한성대학교 교내연구비의 지원으로 이루어졌음.

도는 매우 낮다. 대부분의 MEMS 기반 모션센서는 물리량을 전압의 형태로 출력하고 있으며, 출력 전압이 의미하는 물리량을 데이터시트 등에 공개하고 있다. 또한 모션센서들이 출력하는 값은 필터링 되고 보정되어야 한다. 특히 지자기 센서와 같은 경우에는 정교한 보정이 필요한 경우가 많다 [4]. 또, 센서 자체의 에러, 측정 시 발생하는 잡음 때문에 모션센서를 통해 측정된 값들에서 의미 있는 정보를 얻어내는데 어려움이 있다. 이 점에 착안하여 본 논문에서는 개발자들이 모션센서 자체를 쉽게 이해할 수 있고, 동시에 센서를 사용하는 프로그램을 효과적으로 작성할 수 있도록 실측 데이터 기반 모션센서 에뮬레이터를 개발한 결과를 기술하고자 한다.

개발된 모션센서 에뮬레이터는 실시간, 또는 저장된 모션센서 데이터에 따른 처리 결과를 각종 그래프, 비디오, 간단한 애니메이션을 통하여 시각적으로 출력한다. 즉, 모션센서 자체의 움직임이 어떻게 수치의 변화로 표현되는지 그래프로 보이고, 그 결과를 인식하여 움직임으로 다시 바뀐 결과를 다양하게 시각적으로 확인할 수 있다. 또, 사용자는 실측되어 저장된 다양한 모션센서 정보 가운데, 사용할 모션센서를 선택하여, 제한된 센서만을 사용했을 때, 변화를 바로 확인할 수 있다. 이러한 실측 센서 데이터는 반복적으로 재생될 수 있어, 실제 센서를 이용한 소프트웨어의 개발에서 가장 어려운 점이었던 동작의 재현성 문제를 해결하였다. 즉, 사용자는 자신의 실제 환경에서 사용하는 모션센서 종류, 조합을 세부적으로 선택함으로써 그에 대한 결과를 각종 그래프, 비디오, 게임을 통하여 시각적으로 체험함으로써, 모션센서에 대한 이해와 센서의 사용법을 학습하도록 한다.

이러한 형태의 실측데이터 기반 모션센서 에뮬레이터는 세계적으로도 최초로 개발된 것이다. 특히 개발된 모션센서 에뮬레이터는 기존의 안드로이드 스마트폰 플랫폼에 제공되던 센서 시뮬레이터와 호환성이 있어, 가상적인 움직임이 아닌 실측데이터를 이용한 모션센서 에뮬레이션이 가능하여, 스마트폰 응용 프로그램 개발자들에게 매우 유용하다.

이 연구의 결과는 네이버의 공개소스 프로젝트 개발자 센터에서 공개 소스 프로젝트 [5]로 등록되어, 모든 소스, 수집된 데이터, 문서가 공개되어 있으며, 이를 이용하여 누구나 모션센서의 데이터 수집, 필터 개발, 애니메이션 추가 등의 작업이 가능하다.

## II. 관련연구

본 연구에서 사용된 MEMS 센서는 가속센서, 지자기 센서, 자이로 센서, GPS 등이다.

가속센서(g 센서라고도 불림)는 공간에서 물체가 움직일 때 변화하는 가속도를 측정하는 센서이다. 통상적인 가속센서 [7]는 지구 중력 가속도  $g$ 를 기준으로  $\pm 2g$ 부터  $\pm 8g$  정도까지를 설정하고, 실제 움직임이 나타내는 가속도를 상대적인 전압 값으로 출력한다. 이 전압은 아날로그 디지털 변환을 거쳐 소프트웨어에서 이용된다. 지구상의 모든 물체는 중력방향의 가속도를 겪고 있기 때문에, 고정된 상태에서  $x, y, z$  3축의 공간 좌표계에서 각 방향의 가속도 측정값의 벡터 합이 중력 가속도가 되도록 초기화되어 있다. 중력을 측정하는 기능은 많은 휴대용 가전 기기에서 기울임 인식을 가능하게 한다. 이 기울임은 화면의 회전을 유도하는 동작으로 사용자 인터페이스에서 이용되는 것이 보통이다.

지자기 센서(compass 라고도 불림)는 자기 북극을 측정하는 나침반을 의미한다. 지자기 센서는 센서가 놓인 공간에서 측정되는 자속 밀도 (Magnetic Flux Density)를 측정하는 것으로 [4]와 같은 전형적인 지자기 센서는 센서가 놓인 공간상의 자속을  $0.6\mu T$ (Tesla,  $1T = 10,000$  Gauss)에서  $1.2\mu T$  단위로,  $300\mu T$ 까지 측정하여 그 결과를  $x, y, z$  각축에 대한 전압 값으로 출력한다. 지구의 자기장의 자속 밀도는 약  $50\mu T$ 에 해당한다.

자이로 센서는 물체의 각속도를 측정한다. 자이로 센서는 정속 회전을 하는 움직임을 측정한다는 측면에서 가속도 센서와는 구분된다. 정속 회전 운동은 센서가 부착된 기기를 들고 천천히 움직이는 동작을 감지할 수 있다는 것을 의미하기 때문에, 섬세한 운동을 감지할 수 있다. [8]과 같은 자이로 센서는 초당 1도의 각속도를  $2.5mV$ 로 측정하여 출력하는 3축 자이로 센서이다.

위에서 언급한 3종류의 MEMS 기반의 모션센서는 최근에 가격이 저렴해져서 스마트폰에 적극적으로 적용되고 있다. 현재 가속도센서는 스마트폰뿐만 아니라 많은 피쳐폰에도 적용되고 있다. 특히 가장 먼저 상용화되고 값이 저렴해진 MEMS 가속센서는 각종 기기의 기본 사용자 인터페이스에 적용되어, 화면의 기울임에 의한 화면 전환, MP3 플레이 중에 기기를 흔들면 임의로 선택된 곳을 재생한다거나 하는 등의 기능을 구현하는데 사용되고 있다. 스마트폰과 함께 각광을 받고 있는 증강현실 응용을 위해 반드시 필요한 지자기 센서는 애플의 아이폰 3GS, 삼성의 갤럭시S 등 스마트폰에 적용되고 있다. 애플의 아이폰 4G에는 자이로센서가 채택되어

이를 활용하는 여러 가지 게임이 앱스토어에 출시되고 있다.

본 연구에서 구현하려는 것과 같은 수준의 기능을 가진 모션센서 에뮬레이터의 구현 사례는 거의 없다. 가장 유사한 형태는 그림 1에 보이는 구글이 스마트폰 플랫폼인 안드로이드 개발 도구와 함께 제공하는 센서 시뮬레이터이다 [6]. 이 시뮬레이터는 그림 왼쪽의 박스를 마우스로 움직여, 안드로이드 스마트폰이 유사한 모양으로 움직이는 것을 가정하고 그에 해당하는 모션센서 값들을 출력하도록 되어 있다. 안드로이드 센서 시뮬레이터는 그 움직임을 바탕으로 가속센서, 지자기 센서 데이터를 만들며, 그 두 가지를 조합하여 방향 데이터 (yaw, pitch, roll)를 만들어 제공한다. 또 부가적으로 온도 센서 데이터도 제공한다. 안드로이드 센서 시뮬레이터는 네트워크 연결을 통해서 안드로이드 시뮬레이터와 연결되며, sensorsimulator 클래스를 이용하여 접근이 가능하다.

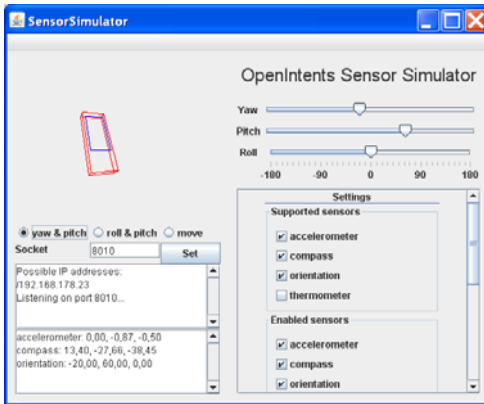


그림 1. 구글의 안드로이드 센서 시뮬레이터 [6]

Fig. 1. Android sensor simulator

안드로이드 센서 시뮬레이터는 모션센서를 이용한 응용 프로그램을 작성하는 경우, 작성하는 소프트웨어가 지정한 움직임에 대하여 정확하게 반응하는지를 테스트하는 목적으로 사용될 수 있으나, 여러 가지 단점 또는 기능 부족 때문에, 제한적으로 이용될 수밖에 없다. 첫 번째 단점은 움직임이 마우스에 의해 만들어지기 때문에, 실제 센서가 만들어 내는 값과 차이가 많다. 실제 모션센서는 다양한 환경 요인에 의해 같은 동작에 대해서도 조금씩은 다른 값을 출력하기도 하며, MEMS 반도체의 한계, 센서가 놓인 위치에 따라, 잡음이 포함된 값을 출력

하는데 반하여, 안드로이드 시뮬레이터는 잡음이 포함되지 않은 거의 완벽한 센서 값을 출력한다. 이는 실제 스마트폰에서의 시험 결과가 개발 단계에서 시뮬레이터를 이용한 경우의 시험 결과와 달라진다는 것을 의미하여, 실제 스마트폰을 이용한 시험을 다시 해야 하는 결과를 가져온다. 두 번째 단점은 센서 동작의 재활용 기능이 없다. 즉, 같은 동작의 재현이 어렵다는 것이다. 소프트웨어 엔지니어들은 사용자의 어떤 동작들에 대하여, 각각 프로그램 내에서의 기능을 정의하고, 그 동작을 인식하는 소프트웨어를 작성하게 되는데, 이 과정은 쉽지 않다. 따라서 반복 시험이 필요하게 되는데, 안드로이드 시뮬레이터는 센서 데이터를 만들려면 매번 마우스로 화면상의 큐브를 움직여야 하기 때문에, 같은 동작을 재생산하는 것이 현실적으로 거의 불가능하다. 또 다른 문제는 방향성 정보를 제외한 센서의 오차나 잡음을 처리하는 필터 소프트웨어를 언제나 스마트폰 응용에서 작성하여, 시험해야 한다는 점이다. 필터링 알고리즘은 매우 플랫폼 독립적이기 때문에, 소프트웨어 작업 생산성이 높은 환경에서 진행되어야 하는데, 안드로이드 SDK가 제공하는 스마트폰 시뮬레이션 환경은 빌드와 실행의 절차가 다소 복잡하며, 실행 결과를 확인하기도 어렵다. 마지막 문제는, 안드로이드 센서 시뮬레이터가 단지 모션센서 데이터만을 출력해 준다는 점이다. 이는 모션센서를 잘 이해하지 못하는 개발자들에게는 어려운 문제이다. 개발자들에게는 모션센서 데이터의 물리량이 의미하는 바를 직관적으로 이해할 수 있는 도구가 필요하다.

본 연구에서 제안한 실측 데이터 기반 모션센서 에뮬레이터 또는 시뮬레이터와는 거리가 있지만, 실측 저장된 센서 데이터를 나중 시점에 분석하여 시각화하는 시스템이 있다. 항공기에 사용되는 블랙박스도 그 일례이다. 블랙박스는 항공기 내의 모든 계기값, 조종 행위, 음성 등을 저장하는 장치로, 저장된 데이터를 바탕으로 특정 시점 (주로 사고가 발생한 시점) 직전의 기체 움직임, 기기 조작 내역, 대화 내용을 복원하고, 그 결과를 시각화하여 재현하는데 이용된다. 최근에는 차량용 블랙박스도 이와 유사한 기능을 수행한다. 차량용 블랙박스는 가속센서 정보와 조향 장치, 제동 장치 등의 기기 조작 내용을 기록하고, 사고가 발생한 순간의 기록을 복원하여 재현한다.

실제 소프트웨어에서 물체의 움직임을 정확하게 감지해내기 위해서는 한 종류의 센서가 아닌 여러 종류의 센서의 값들을 측정하고 잡음을 제거하는

필터링을 거친 후에 움직임을 추출하는 과정이 필요하다. 센서에서 측정된 값을 이용하여 동작인식, 제어 등을 하는 경우, 측정치가 얼마나 실제 움직임을 정확하게 반영하는가에 의해 인식이나 제어의 결과가 좌우된다고 할 수 있다. 그러나 센서가 놓인 환경이 기계적인 움직임이 많거나, 전기적인 노이즈가 많은 경우라면 센서에 의해서 전기적인 신호로 변환된 값들은 노이즈 등의 영향을 받아 실제 값에 원하지 않는 신호가 부가되어 시스템에 전달된다. 또 이런 노이즈는 MEMS 센서 자체에서 발생하기도 한다. 이러한 노이즈 제거 과정, 즉, 관찰 대상인 센서 데이터를 측정된 전체 데이터로부터 걸러내는 과정을 필터링(filtering)이라고 한다. 워낙 MEMS 센서의 출력은 전압의 형태로 나오기 때문에 이러한 필터는 인덕턴스 L과 커패시터 C를 이용하여 하드웨어로 만들 수도 있으나, 부품 값의 상승, 파라미터 수정의 어려움 등의 이유로, 이를 소프트웨어적으로 구현한 디지털 필터 [9]가 통상적으로 사용된다. 디지털 필터는 매우 다양한 종류가 있으며, 필터에 따라 그 필터 특성, 과도 특성, 응답 지연 등이 달라서, 입력 신호, 관찰하려는 신호의 특성에 따라 선택하여 적용한다.

### III. 설계 및 구현

#### 1. 모션센서 에뮬레이터 시스템 설계

본 연구에서 설계 및 구현된 모션센서 에뮬레이터 구조는 그림 2와 같다.

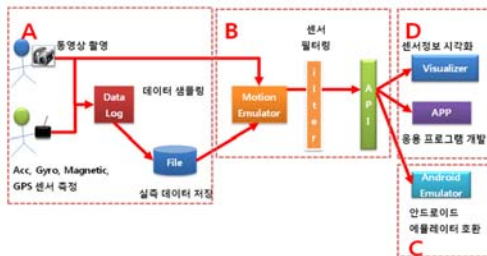


그림 2. 모션센서 에뮬레이터의 구조  
Fig. 2. Motion sensor emulator system

그림의 A 부분은 실제 모션센서를 이용하여 데이터를 수집 저장하는 기능을 수행한다. B 부분은 모션센서 에뮬레이터로 실제 센서나, 저장된 센서

데이터에서 센서 종류를 선택하고, 필터의 종류와 사용 여부에 따라 필터를 거쳐서 네트워크 채널로 데이터를 전송해준다. C는 스마트폰 응용 개발자들을 위하여 관련연구에서 언급되었던 안드로이드 센서 시뮬레이터와 호환성을 유지해주는 변환 모듈이다. D는 응용 프로그램으로 센서 데이터를 이용하는 부분이다. 응용 프로그램의 하나로 이 연구에서는 비주얼라이저를 개발하였다.

#### 2. 센서 데이터의 수집 모듈

본 연구에서는 각종 모션센서를 이용하여 실측 데이터를 수집하였다. 이 실측 데이터를 얻기 위해서 표 1에 기술된 다양한 센서를 이용하였다. 실측에 다양한 모션센서를 이용한 이유는, 스마트폰 이외에서는, 임베디드 기기에 조립된 상태의 모션센서로부터 바로 데이터를 추출할 수 있는 방법이 없었기 때문이다. 또, 실제 모션센서 디바이스의 출력 전압을 그대로 A/D (Analog to Digital) 변환하여 출력하는 센서(Raw)와, 센서 데이터를 필터링하여 노이즈 제거를 하는 센서 (Raw/Filtered), 그리고 보정을 한 상태의 값을 출력하는 AHRS (Attitude and Heading Reference Systems, 항공기에서 자세 및 방향을 측정하는 장치)를 사용하여 센서 (Raw/Filtered/Calibrated)들의 차이를 확인하고, 같은 동작에 대한 다양한 센서의 측정 데이터를 확보하기 위함이다. 특히 AHRS는 센서 데이터가 보정된 후, 센서 모듈 내부에서의 프로세싱을 거쳐 yaw/pitch/roll과 같은 방향 정보를 같이 제공한다. 아이폰과 안드로이드 폰은 센서 데이터를 추출하기 위한 응용 프로그램을 작성하여 데이터를 수집한 뒤, 파일을 전송하여 모션센서 에뮬레이터에서 이용할 수 있도록 하였다.

표 1. 구현에 사용된 모션센서  
Table 1. Motion sensors used

센서 종류	기능	출력 데이터 형태
Phidget[10]	가속도 3축	Raw(R)
Motion Node[11]	지자기 3축	Raw/Filtered(F)
CruizCore(AHRS)[12]	자이로 3축	R/F/Calibrated
iPhone 3GS	가속도 3축	Calibrated
모토로이	지자기 3축	Raw

이 외에도 시중에서 판매되는 USB 형태의 GPS 로거, GPS 수신기를 이용하여 차량에서의 움직임을 측정할 때, 위치 정보도 같이 수집하였다. 또, 비디

오카메라로 모션센서로부터 데이터가 수집되는 상황을 촬영하여, 수집된 데이터에 대한 이해를 높였다. 이 촬영된 영상은 비주얼라이저에서 센서 데이터, 센서 데이터를 이용한 움직임 등과 동시에 재생된다.

모션센서 에뮬레이터의 사용자는 센서의 값을 읽기 위한 디바이스 드라이버 또는 센서 장비와 함께 제공되는 SDK를 이용하는 경우, 자신이 가지고 있는 모션 센서를 이용하여 샘플링을 직접 수행할 수도 있으며, 모션 센서가 없는 경우, 안드로이드 스마트폰이나 아이폰에 샘플링용 앱을 다운로드 받아 스마트폰을 이용한 모션센서 데이터 샘플링을 할 수 있다.

각종 센서를 통해 수집된 데이터는 기본적으로 텍스트 형식으로 저장된다. 저장되는 정보는 가속센서, 자이로, 지자기 센서 각각의 3축 데이터, GPS 위도 경도, 부가정보, 방향 정보 등 이다. 파일 상에서의 모션센서 데이터 저장의 순서는 다음과 같다.

*가속도센서X/가속도센서Y/가속도센서Z/자이로센서X/자이로센서Y/자이로센서Z/지자기센서X/지자기센서Y/지자기센서Z/GPS위도/GPS경도/0/기타추가정보/yaw/pitch/roll*

또, 센서의 측정 데이터만을 가지고 있는 텍스트 파일과 별도로, XML 형식의 메타데이터 파일을 별도로 두어, 사용되는 센서 정보, 샘플링 속도, 동영상 정보를 저장하여, 센서 에뮬레이터에서 이용할 수 있도록 하고 있다. 이를 위해 센서 에뮬레이터는 센서 데이터 수집 상황과, 동영상, 수집된 데이터를 이용하여 XML 메타데이터를 구성하는 기능을 가지고 있다. 구현된 모션 센서 에뮬레이터는 세 종류의 모션 센서를 기본적으로 사용하고 있으나, 전혀 다른 새로운 종류의 센서가 있다면, 프로그램 소스의 수정없이 XML 메타데이터를 수정하는 방법으로 샘플링된 센서 데이터 파일을 이용할 수 있다.

### 3. 모션센서 에뮬레이터

모션 에뮬레이터는 실제 센서장비가 연결된 경우 실시간으로 실제 센서장비의 데이터를 모션센서 값을 필요로 하는 곳으로 소켓통신을 통하여 전송하여 줄 수 있고, 실시간으로 값을 전송하지 않는 경우에는 미리 저장한 모션센서 데이터 파일을 읽어서 모션센서 값을 전송하여 주는 기능을 수행한다. 모션 에뮬레이터의 사용자 인터페이스는 그림 3과 같다.

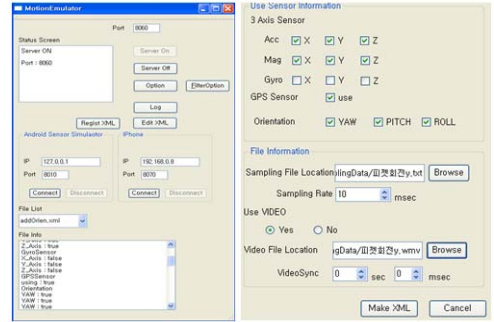


그림 3. 모션센서 에뮬레이터 사용자 인터페이스

Fig. 3. UI of motion sensor emulator

그림 3의 왼쪽에는 수집된 센서 데이터 파일을 선택하고, 이를 전송하기 위한 네트워크 연결 정보를 입력하는 부분이 있으며, 오른쪽은 센서의 종류를 선택하는 화면과, 센서 데이터 수집 후 XML 메타데이터 파일을 생성하기 위한 화면이다.

에뮬레이터에서는 실제 수집된 센서 데이터가 가속, 자이로, 지자기 3가지 센서 각각 3축의 정보를 모두 포함하고 있더라도, 그 중의 일부만 에뮬레이터가 출력하도록 설정할 수 있다. 이는 실제 임베디드 시스템에서는 원가 절감을 위하여 최소한의 센서 디바이스만을 이용하는데, 에뮬레이터에서도 측정된 데이터에서 만들려는 장치에 사용된 센서만을 골라서 출력함으로써, 소프트웨어 개발을 같은 상황에서 할 수 있도록 하기 위함이다.

또 모션센서 에뮬레이터에서는 각 센서별로 필터를 적용할 수 있도록 하였다. 이는 수집된 데이터를 응용 프로그램으로 전달하기 전에 에뮬레이터에서 디지털필터링을 수행한다는 것을 의미한다. 디지털필터는 센서를 이용하는 응용 프로그램에 거의 없이 적용되어야 하는데, 에뮬레이터의 필터 기능을 이용하여, 미리 필터의 유용성을 확인할 수 있다.

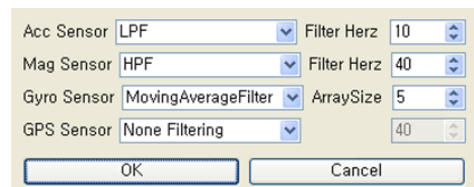


그림 4. 디지털필터의 적용

Fig. 4. Digital filters

그림 4에서 보는 바와 같이 필터는 각 센서 별

로 여러 필터 가운데 하나를 선택할 수 있도록 되어 있고, 각 필터의 세부 특성도 조정할 수 있다.

현재 구현된 모션센서 에뮬레이터에는 저역 통과 필터, 고역 통과 필터, 이동 평균 필터가 내장되어 있다. 또 공개되어 있는 모션 에뮬레이터 소스에는 새로운 디지털 필터를 추가할 수 있도록 API가 정의되어 있다.

① 저역통과필터 (LPF, Low Pass Filter) : 컷 오프 주파수보다 아래 주파수대를 통과하기 쉬운 특성을 가진다.

② 고역통과필터 (HPF, High Pass Filter) : 저역통과필터와 반대로 저주파수 성분을 막고, 고주파수 성분만을 통과시키는 필터이다. 이 필터는 교류와 같은 진동성분을 검출하는 경우, 측정오차 등에 의한 직류성분이나 저주파수 성분이 들어있는 경우 이러한 성분을 제거하기 위하여 사용한다.

③ 이동평균필터 (MAF, Moving Average Filter) : 긴 시간에 얻은 값들의 평균값을 출력하여, 노이즈 등이 서로 상쇄되어 조금 더 실제 값에 가까워지도록 만드는 필터이다.

**4. 안드로이드 센서 시뮬레이터 호환 모듈**

모션센서 에뮬레이터는 실측데이터를 반복 재생함으로써 모션센서를 이용한 동작을 인식하여 게임이나 기타 소프트웨어의 특정 동작을 트리거하는 알고리즘의 개발을 매우 용이하게 한다. 특히 다양한 모션센서가 내장되어 있는 스마트폰 응용 소프트웨어 개발에 있어서 이 기능은 매우 절실하다.

현재 안드로이드 개발 환경에서 제공되는 모션 시뮬레이터는 그 기능적 제약 때문에 실제 센서와는 많은 차이가 나는 데이터를 만들어 낸다. 그 차이는 실제 모션센서에서 항상 관찰되는 잡음이나, 센서의 방향 등이 잘 반영되지 않기 때문이다.

본 연구에서는 스마트폰, 특히 안드로이드 폰을 위한 응용 프로그램 개발자들이 실측된 모션센서 데이터를 이용하여 동작 인식을 활용하는 응용 프로그램을 작성할 수 있도록 기존의 안드로이드 모션 시뮬레이터와의 호환성을 확보하기 위한 기능을 추가하였다. 이 기능을 이용하여 응용 개발자들은 안드로이드 모션 시뮬레이터를 이용하던 것과 완전히 같은 코드로, 어떤 동작을 모션센서로 실측하여 저장한 뒤, 에뮬레이터를 이용하여 반복 재생하여 스마트폰에서의 동작 인식 기능을 효과적으로 개발할 수 있다. 또, 개발된 에뮬레이터의 필터링 기능을 우선 이용하여 필터링의 효과를 검증한 뒤에 필요한 경우, 자신의 소프트웨어에도 필터를 적용할

수도 있을 것이다.

이 안드로이드 호환 모듈은 그림 5에서와 같이 안드로이드 센서 시뮬레이터가 모션센서 에뮬레이터로부터의 데이터를 네트워크 통신을 이용하여 수신한 뒤, 응용 프로그램으로 전달하는 방식으로 구현하였다.



그림 5. 안드로이드 센서 시뮬레이터 기능  
Fig. 5. Android sensor simulator function

**5. 비주얼라이저**

모션센서 에뮬레이터의 응용 프로그램에 해당하는 비주얼라이저는 모션센서의 데이터를 처리한 뒤, 그 결과를 시각화하여 화면에 다양한 방법으로 보여주는 소프트웨어이다. 이 비주얼라이저는 다양한 모션센서를 이용하여 어떤 움직임을 인식해낼 수 있는지를 교육하기 위한 목적으로 작성되었다. 이 기능은 모션센서를 처음 이용하는 소프트웨어 개발자들이 특정 모션센서가 측정해 전송하는 데이터의 의미와, 그 데이터를 처리해서 만들어 낼 수 있는 동작이 무엇인지를 이해하지 못해서 소요되는 시간과 노력을 줄일 수 있도록 도와준다.

구현된 비주얼라이저는 그림 6과 같다.

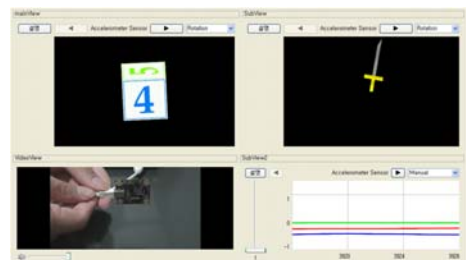


그림 6. 비주얼라이저  
Fig. 6. Visualizer

그림에 보이는 바와 같이 현재 구현된 내용은 모션센서 실측 과정에서 촬영된 동영상, 센서 데이



터를 처리해서 큐브 움직임을 구현한 화면, 칼을 설정하여 센서의 움직임을 게임 같은 소프트웨어에서 칼에 적용한다고 할 때, 구현되는 칼의 움직임에 해당하는 실시간 애니메이션, 그리고 센서 데이터 그래프를 표시하고 있다.

모션인식 소프트웨어 개발자들은 비주얼라이저를 통해서 실제 움직임이 큐브나, 애니메이션에 어떻게 반영되는지 시각적으로 확인할 수 있으며, 모션센서 에뮬레이터에서 각 센서를 선택하거나, 선택하지 않음으로써 각 센서가 동작 인식에 미치는 영향도 바로 확인할 수 있다. 이는 임베디드 시스템에서 최소한의 센서 하드웨어로 원하는 동작 인식이 가능한지를 확인할 수 있도록 한다. 또, 모션센서 에뮬레이터의 필터링 기능을 이용하여 필터가 동작 인식에 미치는 영향도 관찰할 수 있다.

**6. 모션 에뮬레이터의 구현과 데이터의 수집**

본 연구에서 구현된 모션 에뮬레이터는 모든 소스와 설계 문서가 [5]에 공개되어 있다. 따라서 비주얼라이저, 필터를 추가하거나 변경할 수 있으며, 모션 에뮬레이터 자체의 기능도 필요에 따라 쉽게 변경할 수 있도록 하였다. 모션 에뮬레이터의 GUI는 Qt 라이브러리를 사용하여, 마이크로소프트 윈도우즈 환경과 리눅스 환경에 모두에서 실행 가능하도록 하였다. 비주얼라이저는 OpenGL을 이용하여 큐브와 칼 애니메이션을 구현하였다.

실제 실측기반 모션센서 에뮬레이터의 가치는 모션 데이터를 수집하고 이용하는데 있다. 연구에서는 다양한 상황에서의 동작, 자동차의 움직임들을 실측하여 데이터를 제공하고 있으며, 대부분은 센서 데이터가 수집되는 동작을 촬영하여 그 동영상과 파일을 센서 데이터와 같이 제공하고 있다.



그림 7. 모션센서 데이터 수집  
Fig. 7. Collecting motion sensor data

데이터의 수집은 그림 7과 같은 형태로 이루어졌다. [5]에는 지금까지 일상생활에서의 동작과 자

동차의 주행에 따른 센서 데이터 등 다양한 모션센서 데이터가 동영상과 함께 수집되어 있으며, 대부분의 동작은 최소 5회 정도의 샘플링을 통하여 같은 동작에 대해서도 실제 관찰되는 다른 데이터를 볼 수 있도록 하였다.

표 2는 [5]에 수집되고 공개된 실측 데이터열 나열한 것이다. 실제 데이터 샘플은 역시 다운로드가 가능한 데이터 수집 프로그램을 이용하여 사용자에 의해서도 가능하며, 데이터 샘플링 때 주의 사항들과 준비하여야 할 사항들도 문서화 되어있다.

표 2. 실측되어 공개된 모션 센서 실측 데이터  
Table 2. Collected motion sensor data

동작 분류		세부 동작
걷기	빠르게 걷기	1걸음, 3걸음, 5걸음
	천천히 걷기	1걸음, 3걸음, 5걸음, 20초걷기
계단	빠르게 오르기	1계단, 3계단, 5계단
	천천히 오르기	1계단, 3계단, 5계단
	빠르게 내려가기	-
뛰기	뒤로 뛰기	-
	앞으로 뛰기	-
생활	제자리 뛰기	가볍게 뛰기, 제자리 높이 뛰기
	밥 먹기	한술 뜨기, 연속 두술 뜨기
	서명 하기	별표 1번, 3번
진동	세수 하기	좌우 방향, 상하 방향
	손 흔들기	손목 중심, 팔꿈치 중심
자동차	생활 가전 진동	냉장고, 선풍기(강, 약), 세탁기탈수
	프린터 진동	한 페이지 출력 (잉크젯)
회전	급가속 (km/h)	0->40, 60->80, 80->100
	급감속 (km/h)	80->60, 100->80
	급정지 (km/h)	40->0
	급차선 변경	1개 차선, 2개 차선
회전	1축 회전	y축 기준 회전

**IV. 결 론**

MEMS 기반의 모션센서가 저렴하게 등장하면서 모션센서는 스마트폰을 포함한 다양한 기기에 점차 많이 사용되고 있다. 이에 따라 모션센서 관련 소프트웨어 개발에 있어 보다 편리하고 효율적인 개발 도구의 중요성이 크게 대두되고 있다. 하지만 모션센서를 이용한 소프트웨어 작성은 결코 쉬운 일이 아니며, 소프트웨어 엔지니어들은 모션센서가 출력

한 수치의 물리적 의미를 이해하지 못하는 경우도 많다. 앞으로도 모션센서의 응용 분야는 더욱 다양해지고 그에 따른 소프트웨어의 수요도 크게 증가할 것이다. 이에 따라 모션센서 관련 소프트웨어 개발에 있어, 모션센서에 대한 소프트웨어 엔지니어들의 이해도를 높이는 것과, 보다 편리하고 효율적인 개발도구의 필요성이 크게 대두되고 있다.

본 연구에서는 실측된 모션센서 데이터를 저장하고 반복 재생 가능한 모션센서 에뮬레이터와 비주얼라이저를 개발하여 스마트폰 프로그램을 비롯한 많은 모션센서 응용 개발자들이 이용할 수 있도록 하였다. 개발된 모션센서 에뮬레이터는 측정, 저장된 모션센서 데이터에서 센서를 선택적으로 적용할 수 있으며, 필터를 거친 값을 이용할 수도 있다. 무엇보다도 같은 동작을 반복적으로 재생할 수 있어, 개발 생산성이 높아진다. 개발된 모션센서 에뮬레이터는 안드로이드 개발 환경에서 제공되는 에뮬레이터와 호환되어 스마트폰 응용 개발에 큰 도움을 줄 수 있다.

본 연구 결과는 공개 소스 소프트웨어로서 문서와 소스가 공개되어 있으며, 모션센서의 데이터 수집, 필터의 개발, 애니메이션 등 시각화 방법의 추가 등이 가능하다. 개발 결과의 완전한 공개를 통하여 커뮤니티 개발자, 사용자들의 에뮬레이터 기능 추가, 성능 향상, 버그 수정 등의 참여를 통해 결과물을 지속적으로 발전시켜 나갈 것이며, 장기적으로는 역시 커뮤니티의 기여를 바탕으로 각종 센서 종류를 이용한 다양한 모션센서 데이터를 수집하여 DB화하고, 동영상과 함께 제공하는 기반을 만들어갈 예정이다.

**참고문헌**

[1] J. M. Bustillo, R. T. Howe, and R. S. Muller, "Surface micromachining for micro electromechanical systems", *Proceedings of the IEEE*, vol. 86, pp. 1552-1574, Aug. 1998.

[2] Johnson, R. Collin. "There's more to MEMS than meets the iPhone", *EE Times*, 2007.

[3] <http://www.azonano.com/news.asp?newsID=4479> (AtoZ of Nanotechnology Homepage), "Worldwide MEMS Systems Market Forecasted to Reach \$72 Billion by 2011".

[4] Yamaha Datasheet, "YAS529 Magnetic field Sensor", Yamaha Co., 2007.

[5] <http://dev.naver.com/projects/motionemu> (개발 결과물 홈페이지)

[6] <http://code.google.com/p/openintents/wiki/SensorSimulator> (안드로이드 센서 시뮬레이터 홈페이지)

[7] Kionix Datasheet, "KXSD9-2050 Tri-axis Digital Accelerometer Specification", Kionix Co., 2009.

[8] Epson Toyocom Datasheet, "XV-8100CB Ultra Miniature Size Gyro Sensor", Epson Toyomo Co., 2007.

[9] A. Antoniou, *Digital Filters: Analysis, Design, and Applications*, New York, NY: McGraw-Hill, 1993.

[10] [http://www.phidgets.com/products.php?category=5&product\\_id=1056](http://www.phidgets.com/products.php?category=5&product_id=1056) (Phidgets 홈페이지), 'PhidetSpatial 3/3/3'.

[11] <http://www.motionnode.com/> (MotionNode 홈페이지), 'MotionNode 3 DOF Inertial Measurement Unit'.

[12] [http://www.cruizcore.com/e\\_p\\_robot03.html](http://www.cruizcore.com/e_p_robot03.html) (Microinfinity 홈페이지), 'CruizCore XG1010'.

**저 자 소 개**

**이 민 석**



1986년 : 서울대학교 컴퓨터공학과 학사.  
 1988년 : 서울대학교 컴퓨터공학과 석사.  
 1995년 : 서울대학교 컴퓨터공학과 박사.

1999~2002년 : (주)팜팜테크 CTO.  
 1995년~현재, 한성대학교 컴퓨터공학과 교수.  
 관심분야 : 임베디드 시스템 및 소프트웨어, 스마트폰 플랫폼, 멀티미디어 파일 시스템, 공개 소스 소프트웨어.

Email : minsuk@hansung.ac.kr