정규논문 (Regular Paper)
방송공학회논문지 제20권 제1호, 2015년 1월 (JBE Vol. 20, No. 1, January 2015)
http://dx.doi.org/10.5909/JBE.2015.20.1.110
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

안드로이드 기반 자연과학 교육용 디지털 앱북 개발을 위한 사용자 상호작용 라이브러리

이 강 운[a], 백 아 림[a], 최 해 철[a]‡

User Interaction Library for Natural Science Education Digital App-Book on Android Platform

Kang-Woon Lee[a], A-Ram Beak[a], and Haechul Choi[a]‡

요 약

디지털 앱북은 기존의 전자책에서 발전한 형태의 멀티미디어 콘텐츠로서 동영상, 소리뿐만 아니라 모바일 기기의 센서를 이용한 다양한 상호작용을 지원할 수 있는 장점이 있다. 모바일 기기의 발전으로 이러한 디지털 앱북의 수요는 폭발적으로 증가하고 있지만, 상호작용을 지원하기 위해서는 프로그래밍 제작 노력이 많이 필요하기 때문에 공급이 그 늘어난 수요를 따라가기 힘든 실정이다. 이러한 문제를 해결하고자, 본 논문은 자연과학 교육용 디지털 앱북의 흥미를 느끼게 해주는 핵심 요소인 사용자-기기간 상호작용 기능들을 라이브러리로 구현하고 검증하였다. 제안 라이브러리는 사용자 동작 인식부, 기기 동작부, 콘텐츠 동작부로 구성되며, 각 부의 명령을 조합하여 다양한 상호작용 함수를 제공한다. 이러한 설계는 코드의 재사용성, 개발자의 쉬운 이해와 활용성, 넓은 확장성을 지원할 수 있다. 구현된 라이브러리는 상용화를 위한 자연과학 교육용 디지털 앱북 콘텐츠 제작에 직접 이용되었으며, 그 결과 코드 사용량을 크게 줄이고 개발 시간을 단축함으로써 제작 효율을 높일 수 있었다.

Abstract

The digital app-book is an advanced form of the electronic book (e-book), which attracts a lot of interests by the help of video, sound, sensors and a variety of interactions. As mobile devices have evolved, the demand of digital app-books is also rising substantially. However, the distribution of digital app-book contents is hard to meet the demand because the digital app-book requires a lot of programming cost for the interaction. To resolve this problem, Was verified and implementation as a library function of the interaction between device and user. The proposed library consists of three parts (user action recognition, device action, and content action) and provides various user-device interaction functions by combining methods of each part, which can support source code reusability, easy understanding and availability, and wide expandibility. The library was used in the development of natural science education app-book contents. As a result, it could reduce a lot of code lines and facilitate more rapid app-book development.

Keyword : E-Book, Digital App-Book, Mobile Application

a) 한밭대학교 정보통신전문대학원 멀티미디어공학과(Multimedia Engineering, Hanbat National University)
‡ Corresponding Author : 최해철(Haechul Choi)
E-mail: choihc@hanbat.ac.kr
Tel: +82-42-821-1149
ORCID: http://orcid.org/0000-0002-7594-0828
※본 연구는 미래창조과학부와 한국연구재단의 신진연구지원사업(NRF-2013 R1A1A1010344) 및 중소기업청과 (사)한국산학연협회의 산학연협력기술개발사업(C0216895)으로 수행된 연구결과임.
· Manuscript received August 19, 2014 Revised October 13, 2014 Accepted Novemver 11, 2014

Ⅰ. 서 론

최근 스마트폰 확산으로 인해 소비자들은 다양한 형태의 모바일 멀티미디어 콘텐츠를 경험할 수 있게 되었다. 모바일 멀티미디어 콘텐츠는 스마트 기기의 대중화로 인해 누구나 이용할 수 있고, 시간이나 장소에 제약 없이 언제 어디서든 즐길 수 있다. 이에 따라 모바일 멀티미디어 콘텐츠는 문화 산업의 일부분으로 자리매김하고 있으며, 관련 산업도 큰 성장세를 보이고 있다. 스마트 기기 기반 대표적인 모바일 멀티미디어 콘텐츠로 전자책이 있다. 전자책 시장의 대표적인 기업인 아마존은 전용 단말기인 킨들을 바탕으로 풍부한 콘텐츠를 확보하면서 시장 형성에 성공하였으며[1], 이후 다양한 안드로이드 및 iOS 기반 기기 등의 출시와 다양한 유통사의 출현으로[2] 전자책은 기존의 종이 도서를 차츰 대체하면서 그 출판 규모가 꾸준히 성장할 것으로 예측된다[3]. 특히 안드로이드 기반 스마트 기기는 세계 스마트 기기 시장에서 가장 높은 점유율을 보이고 있으며, 오픈소스 정책에 따라 그 개방성과 확장성이 높다. 구글은 안드로이드 운영체제에 포함되어 있는 어플리케이션 마켓인 Google Play를 통해 전자책을 보급하여 전자책 시장을 활성화하고 있다. 애플은 아이폰과 아이패드를 기반으로 App Store, iBooks Author, iBooks Store를 통하여 전자책 시장에 참여하고 있다[4].

한편, 디지털 앱북(App-Book)은 전자책으로부터 발전된 형태의 스마트 기기 콘텐츠이다. 기존의 전자책은 단순히 종이책을 디지털화하여 책의 내용을 보여주는 기능에만 초점이 맞추어져 있었다. 반면 디지털 앱북은 동영상, 소리 등 사용자의 흥미를 유발할 수 있는 콘텐츠들을 담을 수 있으며, 특히 스마트 기기의 센서 등을 이용한 사용자와의 상호작용을 지원하기 때문에 콘텐츠의 표현 방법이 다양하고 콘텐츠의 구성도 폭넓은 설정이 가능하다. 초기 디지털 앱북은 하나의 어플리케이션에 하나의 콘텐츠를 담았지만, 최근에는 여러 권을 통합하여 기획하고 하나의 브랜드 어플리케이션으로 구성하는 추세이다. 이렇듯 콘텐츠 유통업체들은 어플리케이션을 디지털 앱북 스토어 형태로 만들어 하나의 브랜드 어플리케이션 안에서 디지털 앱북을 구매하고 읽는 것이 가능한 새로운 디지털 앱북 콘텐츠 시장을 만들어가고 있다.

이러한 전자책 시장의 성장과 디지털 앱북의 등장으로 다양한 소비 계층에서 콘텐츠의 수요가 급속도로 증가하고 있지만, 콘텐츠 소비 속도에 비해 그 공급량이 매우 부족한 실정이다. 이는 디지털 앱북이 책의 내용을 단순히 디지털화하는 기존의 전자책과는 달리 다양한 상호작용을 지원하기 위한 프로그램 제작 노력과 시간이 상당히 필요하기 때문이다. 이러한 문제점을 해결하고자 본 논문에서는 쉽게 사용 가능한 디지털 앱북 개발용 라이브러리를 제안한다. 제안 라이브러리에서는 사용자 동작 인식, 기기 동작, 콘텐츠 동작으로 구분하여 메소드를 정의하고, 이 메소드들을 조합하여 다양한 상호작용 함수를 제공함으로써 디지털 앱북 개발에서 코드 재사용성, 제작자의 쉬운 이해와 활용성, 라이브러리의 넓은 확장성을 지원하고자 한다. 상업용 자연 과학 교육용 디지털 앱북 개발에 구현된 제안 라이브러리를 직접 활용함으로써 코드의 라인 수 및 개발 시간을 대폭 감소시킬 수 있었다.

본 논문의 구성은 다음과 같다. 제 Ⅱ장에서는 안드로이드 플랫폼, 전자책과 디지털 앱북, 그리고 상호작용에 대해 분석하고, 제 Ⅲ장에서는 본 논문에서 제안하는 디지털 앱북 상호작용 라이브러리에 대해 소개한다. 제 Ⅳ장에서는 본 논문에서 제안하는 라이브러리의 구현 및 검증 결과를 설명하고 마지막으로 제 Ⅴ장에서는 논문의 결론을 기술한다.

Ⅱ. 관련 연구

1. 동적 라이브러리

여러 프로그램이 공통으로 필요로 하는 기능을 별도의 파일로 만든 것을 라이브러리라 한다. 라이브러리는 프로그램 내부에 메소드를 추가하여 사용하는 정적 라이브러리 방식과 프로그램의 필요에 따라 메소드를 호출하는 동적 라이브러리 방식이 존재한다. 라이브러리를 활용하지 않는 환경에서는 동일한 작업을 수행하는 코드가 중복적으로 작성되어 메모리 낭비가 발생할 수 있다. 또한 동일한 수정을

여러 번 반복해야 하는 불편함이 생긴다. 이러한 문제를 개선하기 위해 코드를 동적 라이브러리로 구성하여 동일한 목적을 위한 메모리 낭비를 줄일 수 있고, 또한 수정이 필요할 때 라이브러리에만 수정을 가함으로써 전체 타겟 프로그램의 수정 적용이 가능하다.

2. 사용자 상호작용

사용자 상호작용 장치(User Interaction Device: UID)를 위한 인터페이스(interface)는 일반적인 키보드, 마우스와 같은 물리적 입력 방식에서 멀티 터치, 손동작 인식, 자이로스코프 센서 활용 등과 같은 다양한 방식으로 발전하고 있다. 특히 과거 출력장치로만 활용되었던 화면이 누르고 문지르는 터치를 입력받을 수 있는 터치스크린으로 발전함으로써 사용자와 다양한 상호작용을 지원할 수 있게 되었다.

최근 이러한 다양한 상호작용들은 스마트 기기뿐 아니라 다양한 제품에서도 인터페이스 형식으로 지원되고 있다. 그림 1은 최근 출시된 AUI(Advanced User Interaction) 인터페이스를 이용한 응용 시나리오의 예를 보여주고 있다.

그림 1. 사용자와 장치간의 향상된 상호작용 인터페이스의 예
Fig. 1. An example of advanced user interaction interface

이러한 인터페이스 방식은 편리한 기능을 제공하지만 제품을 개발하는 회사마다 서로 다른 인터페이스 형식으로 정의하고 있기 때문에 다른 회사의 제품들 사이에서 자연스러운 상호연동(inter-operation)이 어렵다. 이에 따라 ISO/IEC JTC1/SC29/WG11 MPEG(Moving Picture Experts Group)에서는 AUI 인터페이스의 데이터 형식을 표준화하기 위해 'MPEG-U part 2: AUI Interface' 프로젝트를 추진하였다[5]. MPEG-U part 2의 목적은 UID에서 발전한 AUI 인터페이스의 데이터 포맷을 표준화하여 장치들 간에 원활한 상호작용 및 상호연동을 지원하는 것이다[6]. 이와 같이 사용자 상호작용은 스마트 기기의 보급으로 인해 급속도로 발전하여 표준화 및 다양한 분야에서 연구되고 있다.

본 논문에서는 스마트 기기 중 가장 많이 사용되고 있는 안드로이드 플랫폼 기반으로, 디지털 앱북 제작 효율을 높이고 디지털 콘텐츠 시장 활성화에 기여하고자 디지털 앱북 상호작용 라이브러리를 설계 및 구현한다.

3. 디지털 앱북

전자책은 종이책의 내용을 디지털화하여 표현하는 콘텐츠와 그 내용을 볼 수 있도록 하는 단말기를 의미한다. 사용자 입장에서 전자책은 기존의 종이책보다 저렴하고, 구매 방법도 온라인 결제로 인해 편리하며, 많은 책을 가벼운 모바일 기기에 담아 이용할 수 있으므로 휴대성도 매우 뛰어나다. 출판사의 입장에서도 책을 출판하는데 필요한 인쇄·제본 작업이 필요하지 않아 제작비가 줄어들고 재고 관리가 필요 없으며, 온라인에서 데이터 형태로 유통이 되기 때문에 물류비도 절약된다. 아직 기기를 별도로 구매해야하는 부담과 종이책에 비하여 콘텐츠가 부족하다는 단점을 지니고 있어 이를 극복하기 위한 단말기 보급의 확대와 콘텐츠 시장의 양적 성장이 요구된다.

디지털 앱북은 기존의 전자책에서 발전한 형태로서, 스마트폰이나 스마트패드와 같은 모바일 기기용 어플리케이션 형태이다. 기존의 정적인 형태의 전자책과는 다른 동적인 형태로써 단순하게 책 내용을 보여주는 역할에 중심을 두는 전자책과 달리, 디지털 앱북은 전자책의 특징을 고스란히 가질 뿐만 아니라 스마트 기기에서 지원하는 다양한 센서 등을 이용함으로써 콘텐츠의 다양한 표현과 구성이 가능하다. 하지만 디지털 앱북 콘텐츠 제작에 있어서 이러한 상호작용을 위한 개발 비용과 시간이 전자책보다 많이 필요하고, 모바일 기기의 운영체제에 따라 별도로 개발해야 한다.

전자책과 디지털 앱북의 성장과 함께 저작권 보호를 위한 보안 기술[7][8], UI/UX를 위한 상호작용의 유형 조사 및 기술 개발[9]-[11], 전자책 시장 활성화를 위한 사용자들의 수

용태도나 구매 행위에 미치는 영향 요인 분석[12]-[15] 등의 관련 연구들이 활발히 이루어지고 있다.

III. 자연과학 교육용 디지털 앱북 상호작용 라이브러리 설계

상호작용은 한쪽에게 일방적으로 정보를 주거나 받는 것이 아니라 서로 주고받는 것이다. 사용자가 스마트 기기의 화면을 터치하거나, 버튼을 누르는 등의 행위가 일어나면 스마트 기기는 전원 on/off, 메시지 보내기, 어플리케이션 실행, 음량 조절 등의 동작으로 사용자의 행위에 대해 반응함으로써 사용자와 기기간의 상호작용을 실현한다.

자연과학 디지털 앱북에서는 다양한 상호작용이 필요하다. 이를 위한 제스처 입력 유형으로는 터치 제스쳐와 모션 제스처로 나뉠 수 있다. 터치 제스처에는 탭, 드래그, 문지르기, 슬라이드, 멀티 터치 등 화면을 터치하는 동작들이 존재한다. 터치에 의한 상호작용은 마우스나 키보드와 같은 별도의 입력 장치를 필요로 하지 않기 때문에 사용자의 행동에 대해 기기가 즉각 반응함으로써 사용자의 즐거움과 호기심을 유발한다. 모션 제스처에서는 자이로스코프 센서를 이용한 기기의 흔들림 정보를 기반으로 하는 틸트, 플릭, 드래그, 플립 등 동작들로 이루어져있다. 이러한 자연과학 디지털 앱북에서의 상호작용은 사용자의 흥미를 유발시켜 콘텐츠에 재미 요소를 추가할 수 있으며, 시간, 공간, 비용 등의 문제로 직접 경험하지 못하는 자연과학 교육과 관련된 다양한 실험이나 현장 체험 등의 간접경험을 제공할 수 있다. 예를 들어 실제로 체험하기 힘든 화석 발굴을 하기 위해서 망치로 돌을 깨고 붓질을 하여 화석을 발견하는 일련의 과정들을 콘텐츠로 담아 낼 수 있다. 사용자가 화면을 탭하여 망치를 움직이면서 암석을 깨는 동작과 화면을 문지르면 붓질하는 동작과 함께 소리를 추가하여 표현함으로써 마치 현장에서 화석을 발굴하는 간접 경험을 한다. 이러한 간접경험을 위해 상호작용들 중 탭과 드래그 동작이 많이 사용되고 있으며 탭의 경우 대부분의 앱북에서 쓰이는 가장 대표적인 동작이다[16]. 더욱 효과적인 경험을 위해서는 가속도 센서를 이용한 확장된 입력 체계, 카메라를 이용한 화상 정보 지원 등 스마트 기기의 각종 센서와 카메라, 스피커 등의 기능들이 이용되어야 한다.

제안하는 자연과학 교육용 디지털 앱북 상호작용 라이브러리는 상호작용과 관련된 기능을 용이하게 개발하고 디지털 앱북 콘텐츠에 새로운 구성을 쉽게 추가 할 수 있도록 정보를 전달하거나 표현하는 주체에 따라 그림 2와 같이 3가지 그룹으로 구성한다. 우선 사용자 동작 인식부는 사용자(디지털 앱북 소비자)가 스마트 기기에 입력을 전달할 수 있는 동작들로 이루어져 있다. 기기 동작부는 스마트 기기의 화면이나 스피커 등을 이용하여 사용자에게 정보를 전

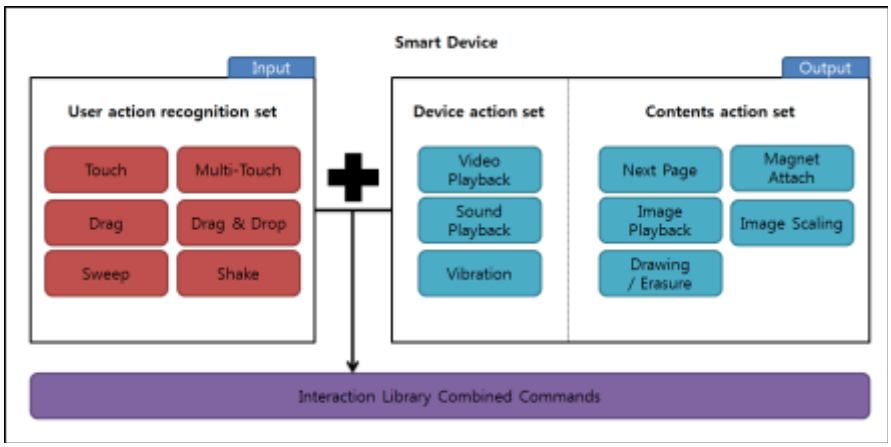


그림 2. 제안 디지털 앱북 상호작용 라이브러리의 구조
Fig. 2. Proposed digital app-book interaction library structure

달하는 그룹이며 진동, 동영상 재생, 소리 재생 등 지속적 동작을 수행하는 명령들로 이루어져 있다. 콘텐츠 동작부는 페이지 이동, 자석 붙기, 이미지 재생, 이미지 확대/축소, 그리기/지우기와 같은 콘텐츠의 표현에 대한 기능을 담당하는 명령들로 이루어져 있다. 제안 상호작용 라이브러리의 구조에서는 각 그룹의 명령이 단독으로 활용될 수도 있고, 그림 2와 같이 각 동작부의 기능을 조합하여 사용자에게 다양한 상호작용 명령을 지원할 수도 있다. 예를 들면 최종 사용자가 한 번의 터치를 했을 때 기기 동작 명령 혹은 콘텐츠 동작 명령 중 제작자가 원하는 명령과 입력 명령을 연결시켜 다양한 상호작용을 구성할 수 있다. 이와 같이 라이브러리를 단순한 명령의 집합으로 제한하지 않고 사용자 동작 인식 명령, 기기 동작 명령, 콘텐츠 동작 명령 체계로 분류하고 이를 유연하게 결합하는 구조로 설계함으로써 콘텐츠 개발자가 쉽게 제안 라이브러리를 이해할 수 있도록 유도하고, 제안 디지털 앱북 라이브러리가 높은 유연성 및 체계성을 갖도록 한다.

제안 상호작용 라이브러리 구조는 디지털 앱북에 자주 사용되는 사용자 상호작용 요소로 구성된 그림 2의 동작 명령뿐만 아니라 추가적인 사용자 동작 인식, 기기 동작, 콘텐츠 동작 명령 및 더 많은 조합 명령도 가능하다. 특히 이 구조는 추후에 새로운 동작 명령을 추가할 때 그것이 사용자 동작, 콘텐츠 동작, 혹은 기기 동작인지를 명확히 구분하여 할당하므로, 특정 그룹 내의 명령들로만 이루어진 조합(예, 사용자 동작 인식 명령만의 조합)을 미연에 방지한다. 만약 사용자 동작 명령끼리만 조합하여 콘텐츠를 제작하면 사용자의 동작을 인식만 할 뿐 기기에서는 아무런 반응이 없을 수 있다. 또한 기기(또는 콘텐츠) 동작 명령끼리 조합하여 콘텐츠를 구성하면 순차적 반응은 수행할 수 있으나 사용자 동작을 정의하지 않았으므로 실행 자체가 불가능할 수 있다. 따라서 두 개 이상 그룹의 명령을 조합시켜 상호작용을 정의하도록 유도함으로써 콘텐츠 개발자가 본 라이브러리를 쉽게 이해하여 제작에 활용할 수 있다. 이와 같이 제안 라이브러리는 높은 확장성과 함께 콘텐츠 제작 오류를 미연에 방지할 수 있는 구조를 갖고 있다. 또한 제안 상호작용 라이브러리는 하나의 상호작용이 하나의 메소드의 형태로 존재한다. 이는 콘텐츠 제작에 활용하기 편리한 형태이며, 무엇보다 높은 재사용성을 갖기 때문에 코드의 중복을 감소시켜 개발과정 효율을 향상 시킨다. 또한 명령 조합이 가능한 구조로 인하여 간결한 코드 구조 형태이기 때문에 라이브러리를 포함한 자연과학 디지털 앱북 프로젝트를 'apk' 확장자 형태의 실행 파일로 변환할 경우 라이브러리가 실행파일에서 차지하는 용량은 약 10Kb로 아주 작은 크기를 지닌다. 어플리케이션 자체 용량의 과도한 증가를 억제하기 위하여 플랫폼 동적 재구성 기법과 [17] 같이 하나의 장치에 다수의 어플리케이션이 라이브러리를 중복 저장하지 않고 일정한 공간에 하나만 저장되도록 하는 기법이 있지만, 라이브러리 구현 이외에 서버 설정, 플랫폼 이미지 등 별도의 작업이 필요하다. 또한, 작은 용량의 라이브러리의 경우, 저장 공간 크기에 대해 고려하지 않아도 된다. 이에 따라 디지털 앱북의 체계적인 개발이 가능하고 유지보수가 편리하며 제작 효율을 크게 높일 수 있다.

다음 각 절에서는 제안 디지털 앱북 라이브러리의 각 그룹에 대해 자세히 설명한다.

1. 사용자 동작 인식부

사용자 동작 인식부는 앞서 기술한 바와 같이 사용자가 스마트 기기에 직접 입력을 전달하는 인터페이스를 담당한다. 사용자 동작은 터치스크린 혹은 자이로스코프 센서를 활용한 입력 방법으로 표 1과 같다. 주로 사용자가 스마트 기기의 화면에 터치하는 유형에 따라 입력 형태 및 정보가 상이하다. 터치 기반의 인터페이스는 X·Y 좌표계에서 화면의 좌측 상단을 원점(0, 0)으로 터치하고 있는 곳의 좌표 및 발생한 이벤트를 제공한다. 대표적인 이벤트로 'ACTION_DOWN', 'ACTION_MOVE', 'ACTION_UP' 등이 있다.

'ACTION_DOWN'은 화면 터치를 시작할 경우 한번만 발생되는 이벤트로서 초기 시작 위치를 좌표로 반환하며 스마트 기기와 사용자의 터치 시작 여부를 알 수 있는 이벤트이다.

'ACTION_MOVE'는 사용자가 스마트 기기를 누르고 있는 동안 터치의 위치가 변경됨으로 인해 발생하는 이벤트로써 이동한 위치의 좌표를 반환한다.

표 1. 사용자 동작 인식 명령
Table 1. User action recognition commands

User Actions	Features	Applicable Functions
Touch	- Basic interface on smart devices like mouse clicks on PC platform	screen contents forward/backward flipping, execution, selection
Multi-Touch	- Operation available on capacitive touch screen devices - Common user experience - Techniques detecting number of fingers pressed and measuring distance between the fingers	Image scaling, image rotation
Drag	- Basic interface on smart devices like mouse drags on PC platform - User experience for page fliping on e-books and digital app-book	screen contents flipping, page flipping
Drag & Drop	- Basic interface on smart devices like moving an icon on PC platform to move an object to another location within screen contents	Item addition, Puzzle
Sweep	- Sweeping action on the smart device screen	Drawing, erasing
Shake	- Exclusive interface supported on smart devices - (x, y, z) coordinate detection using gyroscope sensor	Mixing

'ACTION_UP'은 화면 터치를 끝낼 경우 한번만 발생되는 이벤트로써 마지막 손가락의 위치를 좌표로 반환하며 사용자의 터치가 종료되었음을 알려주는 이벤트이다.

위의 기술한 이벤트는 한 손가락을 터치할 경우 발생한다. 멀티 터치를 구현하기 위해서는 두 번째 손가락에 대한 이벤트가 필요하다. 'ACTION_POINTER_DOWN' 이벤트로 두 번째 손가락과 스마트 기기와의 터치 시작여부를 판단할 수 있으며, 'ACTION_POINTER_UP' 이벤트로는 두 번째 손가락과 스마트 기기와의 터치 종료를 알 수 있다.

터치스크린을 활용한 입력 외의 다른 방법으로 스마트 기기의 움직임을 인식하는 자이로스코프 센서를 이용한 흔들기 동작 인식 기능이 있다. 자이로스코프 센서는 공간상에서 세 가지 축(X, Y, Z)의 기울기 값을 제공함으로써 사용자가 스마트 기기를 돌리고 흔드는 등의 동작 입력 판별에 사용한다. 예를 들어 레이싱 게임에서 자동차의 스티어링 휠 조작과 같이 스마트 기기의 기울기 변화로 사용자의 동작 의도를 파악하여 입력 신호를 줌으로써 게임 내에서 자동차 움직임을 제어할 수 있다.

2. 기기 동작부

기기 동작부는 사용자 동작 인식부와 연동 가능한 명령들 중 진동 모터, 화면, 스피커 등의 장치들을 이용한 진동, 동영상 재생, 소리 재생 등과 관련된 명령들로 구성한다. 이러한 명령은 콘텐츠의 재생 시간이나 사용자 또는 개발자가 설정한 시간 동안 지속적으로 동작하도록 정의되어야

표 2. 안드로이드에서 지원하는 미디어 포맷 및 코덱 목록[18]
Table 2. Media format and CODEC list supported on Android[18]

Types	CODEC	Supported File/Container Formats
Video	H.263	• 3GPP (.3gp) • MPEG-4 (.mp4)
	H.264 AVC	• 3GPP (.3gp) • MPEG-4 (.mp4) • MPEG-TS (.ts)
	MPEG-4 SP	• 3GPP (.3gp)
	VP8	• WebM (.webm) • Matroska(.mkv)
Audio	AAC LC	• 3GPP (.3gp) • MPEG-4 (.mp4, .m4a) • ADTS raw AAC (.aac) • MPEG-TS (.ts)
	HE-AACv1	
	HE-AACv2	
	AAC ELD	
	AMR-NB	• 3GPP (.3gp)
	AMR-WB	
	FLAC	• FLAC (.flac) only
	MP3	• MP3 (.mp3)
	MIDI	• Type 0 & 1 (.mid, .xmf, .mxmf) • RTTTL/RTX (.rttl, .rtx) • OTA (.ota) • iMelody (.imy)
	Vorbis	• Ogg (.ogg) • Matroska (.mkv)
	PCM/WAVE	• WAVE (.wav)

한다. 구체적으로 진동관련 명령어는 스마트 기기의 진동 모터를 위한 초단위의 시간 설정 기능을 포함해야 하며, 동영상과 소리 관련 명령어는 기기에 저장되어 있거나 인터넷으로 연결된 동영상 또는 사운드 파일의 제어를 담당해야 한다. 표 2는 안드로이드 플랫폼에서 재생 가능한 코덱과 파일 포맷이다. 이러한 유형의 파일들을 이용하여 'VideoView'를 이용한 동영상 재생, 'MediaPlayer'를 이용한 소리 재생 동작을 하게 된다.

3. 콘텐츠 동작부

콘텐츠 동작부는 사용자 동작 인식부와 연동 가능한 명령들 중 사용자 동작에 즉시 반응하여 콘텐츠에 변화를 주는 명령들로 구성한다. 이는 사용자 동작 인식부 명령과 조합하여 사용할 수 있으며 페이지 이동, 자석 붙기, 이미지 확대/축소, 그리기 및 지우기, 이미지 재생 동작 등을 지원한다.

페이지 이동 동작 명령은 현재 액티비티(activity)에서 다른 액티비티로 전환을 지시한다. 디지털 앱북뿐만 아니라 대부분의 어플리케이션에서 자주 쓰이는 기능이다. 레이아웃(layout) 상에서 버튼을 하나 생성하여 위치를 지정하고 'Intent' 클래스를 이용한 액티비티 이동을 지시할 수 있다.

자석 붙기 동작 명령은 영역좌표를 지정하여 해당 영역 안에 오브젝트가 이동할 경우 미리 지정된 위치로 오브젝트를 이동시키고 이외의 영역일 경우 오브젝트를 원위치로 되돌린다. 이 명령은 오브젝트의 움직임을 정형화하여 모든 사용자에게 동일한 오브젝트의 이동 경험을 제공할 수 있으며, 사용자가 미리 의도된 콘텐츠의 구성 시나리오를 자연스럽게 따라가도록 유도할 수 있다.

이미지 확대/축소 동작 명령은 동작할 대상 오브젝트를 지정한 후 입력받는 값에 따라 이미지 중심점을 기준으로 스케일 변환을 적용한다.

그리기 및 지우기 동작 명령은 그리기용 메소드와 지우기용 메소드로 구성된다. 메소드들은 'View' 클래스를 상속 받은 커스텀 이너 클래스(custom inner class)에 포함된다. 그림판 역할의 레이아웃 위에 해당 커스텀 이너 클래스를 올리면 해당 레이아웃은 선을 그리거나 지울 수 있는 기능을 할 수 있다. 커스텀 이너 클래스는 리스너(Listener)도 포함하고 있어 펜 사이즈, 펜 색상, 그리기 또는 지우기 모드 설정 등의 매개변수를 필요로 한다.

이미지 재생 동작 명령은 이미지가 표시 될 'ImageView'가 초기 상태를 숨김으로 설정하고 재생을 시작하면 숨김 상태를 해제하여 스마트 기기의 화면에 이미지를 보여준다.

IV. 구 현

제안 라이브러리의 구현 환경은 다음과 같다. 개발 언어는 JAVA이며, 개발 툴은 Eclipse 4.2.1, ADT v22.3.0 플러그-인을 사용하였다. 그리고 운영체제는 안드로이드 SDK 4.1.2 버전에 기반을 두었다. 구현된 라이브러리는 상용화 예정인 자연과학 교육용 디지털 앱북 콘텐츠에 직접 사용되어 그 효율성을 검증하였다.

1. 상호작용 라이브러리 클래스

상호작용 라이브러리 클래스는 안드로이드 프로젝트로 구현하였다. 제안 상호작용 라이브러리는 기기 동작부와 콘텐츠 동작부 각각의 명령을 포함한 독립적 메소드와 사용자 동작 인식부와 콘텐츠 동작부 혹은 기기 동작부의 명령들을 결합한 조합 명령 메소드가 'Interaction_lib'라는 클래스로 구성되어 있다. 이 클래스에는 기기 동작부 3개 메소드, 콘텐츠 동작부 5개 메소드, 사용자 동작 인식부 6개 메소드가 있으며, 사용자 동작 인식 메소드 하나는 기기 동작부 또는 콘텐츠 동작부의 모든 메소드와 결합이 가능하기 때문에 총 48가지 조합의 상호작용이 가능하다.

상호작용 명령들을 디지털 앱북 콘텐츠 개발에서 라이브러리로 올바르게 활용하기 위해서는 라이브러리 개발 시 프로젝트의 설정에서 'Is Library' 항목을 체크함으로써 라이브러리로 선언해야 한다. 이와 같이 수행하면 디지털 앱북 개발을 위한 프로젝트에서 제안 상호작용 라이브러리를 추가하여 자유롭게 활용할 수 있다.

제안 상호작용 라이브러리의 기기 동작부 및 콘텐츠 동작부 명령 하나는 하나의 메소드로 이루어져 있고 사용자

동작 인식 명령은 사용자의 동작 판별 기능 안에 기기 동작 혹은 콘텐츠 동작 명령 메소드를 호출하는 형태로 구현하였다. '드래그&드롭'에 의한 자석 붙기 상호작용을 예로 들면 그림 3과 같이 'attach'라는 콘텐츠 동작 메소드가 존재하고 사용자 동작 인식 메소드 'drop_attach'에서 'attach'를 호출한다. 이와 같은 방식으로 기기 동작이나 콘텐츠 동작 메소드를 사용자 동작 인식 메소드 내부에 위치시켜 하나의 상호작용 명령을 구성한다.

a) Magnet Attach(Contents action set)

```
public void attach(View v, ImageView dst_iv){
  v.setX(dst_iv.getX());
  v.setY(dst_iv.getY());
}
```

b) Drag & Drop(User action recognition set)
+ Magnet Attach(Contents action set)
= Interaction command(drop_attach)

```
public void drop_attach(ImageView drag_iv, final ImageView
dst_iv, final int init_x, final int init_y){
  drag_iv.setOnTouchListener(new View.OnTouchListener() {
  @Override
  public boolean onTouch(View v, MotionEvent event) {
    int touchX = (int) event.getRawX();
    int touchY = (int) event.getRawY();
    switch (event.getAction()) {
      case MotionEvent.ACTION_MOVE:
        v.setX(touchX - (v.getWidth() / 2));
        v.setY(touchY - (v.getHeight()));
      break;
      case MotionEvent..ACTION_UP:
        if (dst_iv.getX() < touchX && touchX < dst_iv.getX()
+ dst_iv.getWidth() && dst_iv.getY() < touchY && touchY <
dst_iv.getY() + dst_iv.getHeight()) {
          attach(v,dst_iv);
        }else{
          v.setX(init_x);
          v.setY(init_y);
        }
      break;
    }
    return true;
  }
 });
}
```

그림 3. 콘텐츠 동작 메소드와 사용자 동작 인식 메소드로 상호작용 메소드를 구성한 예
Fig. 3. Example of an interaction method including content operation and user action methods

2. 디지털 앱북 콘텐츠 응용

제안 라이브러리의 패키지 구조는 안드로이드 프로젝트의 기본구조를 따르고 있으며, 디지털 앱북 콘텐츠 개발과정에서 제안 라이브러리를 적용시키기 위해서는 라이브러리 프로젝트 목록에서 'Interaction_lib' 라이브러리 프로젝트를 자연과학 교육용 디지털 앱북 프로젝트에 추가하면 된다. 이와 같은 방법으로 다양한 디지털 앱북 솔루션에서 편리하게 사용자 상호작용 라이브러리를 활용할 수 있다.

제안 디지털 앱북 개발을 위한 상호작용 라이브러리의 효율성을 검증하기 위하여 자연 과학 교육을 위한 '지층과 화석' 디지털 앱북과 '식물' 디지털 앱북 개발에서 제안 상호작용 라이브러리를 활용하였다. 완성된 '지층과 화석'과 '식물' 콘텐츠에서 사용자 동작 인식부 명령과 기기 동작부 또는 콘텐츠 동작부 명령을 조합한 사용자 상호작용이 적용된 예를 사용자 동작 인식부 명령 6가지를 기준으로 그림 8에 나타내었다. 사용자 동작 인식부는 사용자가 스마트 기기와 상호작용을 위해 필요한 입력 명령으로 구성되어 있고, 기기 동작부와 콘텐츠 동작부 명령들은 입력에 대해 사용자에게 출력하는 명령으로 구성하였다.

'touch_page' 상호작용 명령은 현재 페이지에서 다음 페이지로 화면을 전환시킨다. 이러한 명령을 실행시키기 위해서는 주로 버튼을 활용하여 사용자의 입력을 받으며, 이미지가 버튼이라는 것을 사용자가 쉽게 인지하도록 버튼에 이펙트를 주는 추가 작업이 필요하다. 제안 라이브러리는 쉬운 조작을 위해 터치에 반응하는 영역을 화면 전체로 지정하여 버튼 활용 없이 페이지 넘김 명령을 실행 시킬 수 있다.

'mt_si' 상호작용 명령은 화면 중앙에 있는 물체가 사용자의 두 손가락 사이의 거리에 따라 돋보기 내부에서 유연하게 확대 혹은 축소가 된다. 사용자에게 실제 돋보기로 물체를 관찰하는 것과 유사한 경험을 제공한다.

'drop_attach' 상호작용 명령은 실험 도구들과 실험 재료들을 상자에 담는 동작을 한다. 자연과학 교육용 디지털 앱북에서 준비물을 확인하는 페이지 등에서 활용하였다. 사용자가 올바른 도구나 재료를 모두 상자 영역에 놓으면 다음 페이지로 진행 할 수 있도록 구현하였다.

'drag_mov' 상호작용 명령은 물체를 드래그를 하면 물체 그림이 바뀌면서 동영상이 재생된다. 자연과학 교육용 디지털 앱북에서는 시추선이 바다위에 자리 잡고 있을 때 시

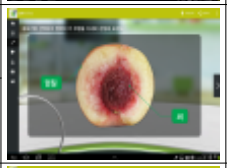
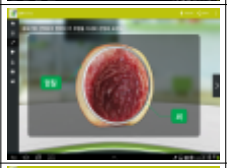
Combined commands	User action recognition commands	Interaction		Interaction description
	Device (or Contents) action commands	Before	After	
touch_page	tap next_page			User flips to next page by touch.
mt_si	multi_touch scailing_img			User enables magnifier by multi-touch.
drop_attach	drag_drop attach			User collects particular objects into a box.
drag_mov	drag play_move			User drags the rig ship from top to bottom in order to activates video play.
shake_img	shake play_img			User shakes the erlenmeyer flask in order to mix some liquids.
sweep_de	sweep draw_erase			User sweeps the top-layer image to find the bottom-layer objects

그림 4. 상호작용을 위한 조합 명령 구현 예
Fig. 4. Combined commands to facilitate interactions

추선 위에 손 모양이 나타나 위에서 아래로 드래그하면 시추선이 내려오면서 동영상이 재생되도록 구현하였다. 이는 콘텐츠 진행을 위해 사용자에게 가이드 역할을 하며, 동영상을 통해 정보를 제공할 수 있다.

'shake_img' 상호작용 명령은 사용자가 스마트 기기를 흔들면 콘텐츠의 내용이 변화하도록 한다. 자연과학 교육용 디지털 앱북에서는 사용자가 스마트 기기를 흔들면 삼각 플라스크 안에 있는 내용물이 점점 붉게 짙어지도록 구현하였다. 구체적으로 자이로스코프 센서에서 각 축의 기울기 값을 측정하여 개발자가 설정한 수치 이상의 기울기 값으로 조건이 만족한다면 붉게 변한 삼각 플라스크 이미지를 재생한다. 색상 변환이 모두 끝나면 자이로스코프 센서로부터의 입력을 중지한다.

(a) Apply Interaction library code

```
drop_attach(drag_iv, dst_iv, 0, 300);
```

(b) Not apply Interaction library code

```
public void drop_attach(ImageView drag_iv, final ImageView
dst_iv, final int init_x, final int init_y){
  drag_iv.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
      int touchX = (int) event.getRawX();
      int touchY = (int) event.getRawY();
      switch (event.getAction()) {
        case MotionEvent.ACTION_MOVE:
          v.setX(touchX - (v.getWidth() / 2));
          v.setY(touchY - (v.getHeight()));
        break;
        case MotionEvent.ACTION_UP:
          if (dst_iv.getX() < touchX && touchX < dst_iv.getX()
+ dst_iv.getWidth() && dst_iv.getY() < touchY && touchY <
dst_iv.getY() + dst_iv.getHeight()) {
            v.setX(dst_iv.getX());
            v.setY(dst_iv.getY());
          }else{
            v.setX(0);
            v.setY(300);
          }
        break;
      }
      return true;
    }
  });
}
```

그림 5. 제안 라이브러리 적용과 미적용 시 상호작용 구현 예
Fig. 5. Example of interaction implementations when the proposed library is used or not used

'sweep_de' 상호작용 명령은 사용자가 화면을 문지른 영역에 아래 레이어의 이미지를 출력하도록 한다. 자연과학 교육용 디지털 앱북에서는 모래를 문지르면 모래에 가려져 있던 화석이 나오도록 구현하였다. 즉, 두 개의 레이어로 구성되어 화면을 문지르게 되면 상단 레이어인 모래 이미지가 사용자 손가락의 이동에 따라 지워지고 하단 레이어인 화석이 표시된다.

그림 5는 'drop_attach' 상호작용의 실제 코드를 나타낸다. 제안 상호작용 라이브러리를 활용하면 그림 5의 (a)와 같이 자석붙기에서 이동할 오브젝트, 이동할 오브젝트가 붙을 대상 오브젝트, 자석붙기를 실패했을 때 돌아갈 초기 x, y 좌표의 입력으로 간단하게 상호작용을 적용시킬 수 있다. 하지만 제안 라이브러리를 활용하지 않고 상호작용을 직접 개발하기 위해서는 안드로이드의 각종 메소드를 직접 이해하여 그림 5의 (b)와 같이 27줄의 코드를 작성해야 한다. 이와 같이 제안 라이브러리는 코드의 라인 수를 대폭 감소시킬 수 있다. 코드 라인 수 측면에서 다른 상호작용에 대한 제안 라이브러리의 효율성을 표 3에 나타내었다. 표 3에서 보이는 것처럼 라이브러리를 활용하지 않는다면 상

표 3. 상호작용 명령의 코드 라인 수
Table 3. Code lines of interaction commands

Interaction command set								
Interaction commands	Number of lines		Interaction commands	Number of lines		Interaction commands	Number of lines	
	Using the proposed library	Original		Using the proposed library	Original		Using the proposed library	Original
tap_mov	1	39	mt_mov	1	80	sweep_mov	1	102
tap_sound	1	19	mt_sound	1	60	sweep_sound	1	82
tap_vibrate	1	12	mt_vibrate	1	53	sweep_vibrate	1	75
tap_page	1	17	mt_page	1	58	sweep_page	1	80
tap_attach	1	10	mt_attach	1	49	sweep_attach	1	71
tap_img	1	14	mt_img	1	54	sweep_img	1	76
tap_si	1	12	mt_si	1	53	sweep_si	1	75
tap_de	3	68	mt_de	3	109	sweep_de	3	131
drag_mov	1	49	drop_mov	1	56	shake_mov	1	62
drag_sound	1	29	drop_sound	1	36	shake_sound	1	42
drag_vibrate	1	22	drop_vibrate	1	29	shake_vibrate	1	35
drag_page	1	27	drop_page	1	34	shake_page	1	40
drag_attach	1	20	drop_attach	1	27	shake_attach	1	33
drag_img	1	24	drop_img	1	31	shake_img	1	37
drag_si	1	22	drop_si	1	29	shake_si	1	35
drag_de	3	78	drop_de	3	85	shake_de	3	91

호작용을 적용시키기 위해서 많은 코드를 작성해야 한다. 더욱이 디지털 앱북 콘텐츠에서는 반복적으로 자주 상호작용을 적용시켜야 하므로 제안 라이브러리가 감소시키는 코드의 양이 훨씬 더 많을 수 있다. 또한 그림 5와 같이 제안 방법은 상호작용 별로 메소드를 정의하였다. 이는 콘텐츠 개발자가 쉽게 이해하면서 상호작용을 적용 시킬 수 있는 개발 편이성을 제공할 수 있다.

V. 결 론

전자책의 발전된 형태인 디지털 앱북은 전자책보다 많은 기능들을 담을 수 있지만 이에 비례하여 많은 제작 시간이 필요하다. 이러한 문제점을 해결하기 위해 본 논문에서는 안드로이드 기반의 디지털 앱북 제작에서 자주 사용되는 상호작용을 분석하여 사용자 동작 인식부, 기기 동작부, 콘텐츠 동작부의 체계로 구성하고 조합 명령으로 상호작용 메소드를 지원하는 방법을 제안하였다. 또한 제안된 라이브러리를 실제 디지털 앱북 제작에 활용하여 콘텐츠를 개발함으로써 그 효율성을 검증하였다. 이와 같은 라이브러리의 개발은 디지털 앱북 제작을 활성화시켜 디지털 앱북 콘텐츠 증가와 시장 성장에 큰 도움을 줄 수 있을 것이다. 향후 과제로 라이브러리에서 더욱 발전시켜 통합된 디지털 앱북 저작도구를 개발한다면 더욱 쉽게 디지털 앱북을 개발할 수 있는 환경을 제공할 수 있을 것이다.

참 고 문 헌 (References)

[1] Ji-Hwan Kim, "Amazon's new business success secret: Leverage Strategy", SERI Management Notes, No. 162, pp.1-10, 2012.

[2] Yang-Hoe Song, Yeong-Sik Kang, "Electronic publications promoting standardization direction for industry competitiveness", TTA Journal, No. 130, pp.14-19, 2010.

[3] Ju-Young Lee, "E-book Market Status and Forecast of the value chain and market structure change in book publishing", KISDI Information Communication and Broadcasting Policy, Vol. 26, No. 8, pp.1-4, 2014.

[4] Yong-Ho Jang, Byung-Hoon Kong, “The emergence and ensuing typology of global ebook platform -The case study on Google eBook, Amazon Kindle, Apple iBooks Store”, Journal of academia-industrial technology Vol. 13, No. 8, pp.3389-3404, 2012.

[5] Min-Hye Kim, Jung-Gon Si, “A Study on the relationship between Touch interactions and Story in Tablet PC App-book : Focused on Children’s App-Story Book”, Journal of the HCI Society of Korea, pp.996-998, 2012.

[6] Gukhee Han, A-Ram Beak, Haechul Choi, “MPEG-U part 2 based Advanced User Interaction Interface System”, Journal of the Korea Contents Association, Vol. 12, No. 12, pp.54-62, 2012.

[7] Hameed, N., Cheah, Y.-N, and Rafie, M., “An e-book Personalization Architecture With Digital Rights and Encryption Procedures”, Information Technology (ITSim), Vol.3, pp.1222-1227, 2010.

[8] Jong-Wook Bae, Seon-Hwan Jeong, "Scale Resilient Watermarking System for E-book Copyright Management", Journal of Information Science, Vol. 19, No. 6, pp.356-360, 2013.

[9] Veronica Liesaputra, Ian H. Witten, and David Bainbridge, "Creating and Reading Realistic Electronic Books", IEEE Computer Society, Vol42, Issue2, pp.72-81, 2009.

[10] Ji-Eun Kwon, Bo-Young Kim, “A Study Funology for Reformatting to App Book: Focused on 'Moo, Baa, La La La!' of App Book”, Journal of Cartoon and Animation Studies, No. 30, pp.221-243, 2013.

[11] So-Young Park, Jea-Hwan Lee, “A study on the preference comparison according to the difference of the user mode of children's app-book interface”, Journal of Digital Design, Vol. 12, No. 4, pp.599-608, 2012.

[12] Jin Gerlach, Peter Buxmann, “Analyzing Electronic Book Acceptance : A Compatibility Perspective”, System Sciences (HICSS), pp.2813-2822, 2013.

[13] Jun-Sik Kim, Ok-Hwa Bae, Mi-Sol Lim, Jae-Won Lee, "A Study on E-book Contents Distribution Business Model and its Adoption Attitude of College Students", Journal of practical engineering education, Vol. 4, No. 2, pp.75-84, 2012.

[14] Hye-Ran Jang, "The Use of Web-based Electronic Books among Undergraduate Students", Journal of Information Management, Vol. 23, No. 4, pp.233-256, 2006.

[15] Jong-Ok Lee, Yong-Moon Kim, "A Study on the Impact of the App-Book Purchasing Behavior of Smart phone Users in Korea", Journal of Korea Electronic Commerce, Vol. 18, No. 3, pp.45-67, 2013.

[16] Kang, Eun-Jin, “Analyzing the Types of Contents and Interactivity of Picture Book Applications, Journal of Children's Literature and Education, Vol. 14, No. 3, pp.1-23, 2013.

[17] Byeong-Seong Park, Yong-Deok Yu, Hoon Choi, “A Dynamic Reconfiguration Method of Android Platform”, Journal of Information Science, Vol. 38, No. 1, pp.16-24, 2011.

[18] Android official web site, accessed Aug, 8, http://developer.android.com/guide/appendix/media-formats.html

저 자 소 개

이 강 운

- 2012년 2월 : 한밭대학교 멀티미디어공학과 (학사)
- 2013년 2월 ~ 현재 : 한밭대학교 정보통신전문대학원 멀티미디어공학과 석사과정
- ORCID : http://orcid.org/0000-0002-2599-3286
- 주관심분야 : 인터랙션, 모바일 프로그래밍

백 아 람

- 2012년 2월 : 한밭대학교 멀티미디어공학과 (학사)
- 2014년 2월 : 한밭대학교 멀티미디어공학과 (석사)
- 2014년 3월 ~ 현재 : 한밭대학교 정보통신전문대학원 멀티미디어공학과 박사과정
- ORCID : http://orcid.org/0000-0001-7773-2347
- 주관심분야 : 모바일 GPU 처리, 모바일 멀티미디어 처리

최 해 철

- 1997년 2월 : 경북대학교 전자공학과 (공학사)
- 1999년 2월 : 한국과학기술원 전기및전자공학과 (공학석사)
- 2004년 8월 : 한국과학기술원 전기및전자공학과 (공학박사)
- 2004년 9월 ~ 2010년 2월 : 한국전자통신연구원 방송미디어연구부 선임연구원
- 2010년 3월 ~ 현재 : 한밭대학교 정보통신공학과 교수
- ORCID : http://orcid.org/0000-0002-7594-0828
- 주관심분야 : 영상 통신, 비디오 부호화, 멀티미디어 처리, 모바일 프로그래밍