

스마트폰 카메라를 이용한 모바일 도서 관리 어플리케이션

정 명 범*, 강 미 애**, 고 일 주*

Mobile Book Management Application using Camera of Smart Phone

Myoung-Beom Chung*, Mi-Ae Kang**, Il-Ju Ko*

요 약

본 논문에서는 스마트 폰의 카메라와 온라인에서 제공되는 Open-API 책 정보를 이용하여 사용자가 쉽게 자신의 도서를 관리할 수 있는 어플리케이션을 제안한다. 스마트 폰의 카메라는 책에 있는 Bar-Code, QR-Code 등을 자동으로 인식 할 수 있으며, 인식한 내용 정보를 바탕으로 Open-API를 이용하여 보다 구체적인 도서 정보를 제공 받을 수 있다. 제안 어플리케이션은 사용자가 제공 받은 책 내용을 스캔 목록이나, 나의 책 리스트에 저장하며 필요시 현재까지 읽은 페이지를 메모하고, 다 읽은 책의 경우 간단한 코멘트를 작성할 수 있다. 즉, 제안 어플리케이션은 사용자가 읽는 책을 효과적으로 관리할 수 있는 유용한 기능을 제공한다.

▶ Keyword : 스마트 폰, 도서 관리, 바코드, QR-Code

Abstract

In this paper, we propose a mobile book management application using the camera of a smart phone and the open API book information provided online. The camera automatically scans the bar-code or QR-code of a particular book. The code of that book is used to transmit detailed information to the open API. Then the proposed application records this information for use in an "Scan list" or an "My book list." This application notes the page of the book that the user is currently reading and, if the user has read the entire book, he or she can comment about it. Therefore, the proposed application provides useful functions so that the users can manage their books effectively.

▶ Keyword : SmartPhone, Book Management, Bar-Code, QR-Code

• 제1저자 : 정명범 • 교신저자 : 고일주
• 투고일 : 2011. 08. 24, 심사일 : 2011. 09. 30, 게재확정일 : 2011. 10. 10.
* 숭실대학교 미디어학부 (Dept. of Media, Soongsil University)
** (주)야긴스텍 (YAGINSTEK Co., Ltd.)

I. 서론

최근 모바일 환경의 발달과 폭 넓은 인터넷의 확산으로 사용자들은 보다 풍부하고 깊이 있는 정보를 손쉽게 접할 수 있게 되었으며, 휴대 단말기 또한 통화 목적의 휴대 전화를 넘어서 손 안의 작은 컴퓨터라 불리는 스마트 폰의 대중화를 이끌어 냈다. 스마트 폰에 있는 카메라도 기존 사진을 찍어 저장하는 용도로만 사용되던 것이 현재에는 증강 현실(AR: Augmented Reality) 게임, AR 내비게이션, Bar-Code와 QR-Code(QR: Quick Response) 인식 등 다양한 형태의 정보 제공 역할을 하고 있다 [1].

현재 스마트 폰의 카메라를 이용한 증강 현실 기능은 마커를 인식하여 그 위치에 원하는 이미지, 애니메이션 등을 표현하는 기술과 GPS를 이용하여 사용자 위치 값을 혼합하고, 특정 위치를 찾아갈 수 있게 하는 내비게이션 기능에 활용된다. 마커 인식을 통한 애니메이션 기술은 카메라 Overlay 위에 OpenGL ES를 사용하여 애니메이션을 나타내며, 주로 게임 분야에 사용 된다 [2, 3]. GPS를 혼합한 내비게이션 기능은 스마트 폰에서 제공되는 Location Manager의 Heading과 Location을 사용한 것으로 iPhone ARkit에서 이 기능을 보다 쉽게 사용할 수 있도록 오픈 소스 형태로 제공하고 있다 [4].

QRDic과 쿠투쿠루 등으로 많이 알려진 Bar-Code와 QR-Code 인식 기술은 코드 내의 가지고 있는 데이터를 스마트 폰의 카메라로 인식하고, 그 데이터를 서버에 전송하여 해당 코드의 구체적인 설명을 웹 페이지 형태로 제공 받는다. 이러한 Code 인식 기술은 기존 컴퓨터 기반과 모바일 기반에서 많이 연구되었던 것으로 [5, 6], 현재에는 아이폰의 iOS 기반에서 손쉽게 사용할 수 있게 Zxing과 Zbar에서 라이브러리 형태로 제공되고 있다 [7, 8]. QR 코드나 Bar 코드 인식 기술은 최근 사회적 이슈가 되고 있는 것 중 하나로써, 2010년 하반기 광고를 하기 위해 QR 코드를 사용하기도 하였으며, 각종 포털 사이트 (네이버 QR코드, Daum 코드 등)에서도 점차 QR 코드를 통한 마케팅이 활발히 이루어지고 있다.

따라서 본 논문에서는 아이폰의 Bar-Code와 QR-Code 인식 기술을 이용하여, 사용자가 쉽게 도서를 관리할 수 있는 모바일 도서 관리 어플리케이션을 제안한다. 제안 어플리케이션은 2009년 Apple의 Mac PC와 Mac Book에서 개인 도서를 관리 할 수 있게 제공된 Books라는 어플리케이션과 비슷한 기능을 하는 것이다. 그러나 Apple 사의 Books 어플리케이션은 컴퓨터와 노트북에서만 사용할 수 있고, Bar-Code만 인식하며, 도서 Database를 사용자가 주기적으로 수동

업데이트해야 하는 문제점이 있다. 반면에 제안 어플리케이션은 사용자가 쉽게 들고 다니는 아이폰에서 사용할 수 있고, Bar-Code와 QR-Code까지 인식 가능하다. 게다가 Naver Open API와 Google 책 API를 사용하기 때문에 도서 Database를 사용자가 업데이트할 필요가 없다는 장점이 있다. 따라서 제안 어플리케이션은 사용자가 도서관, 서점, 자신의 집 등 어느 위치에 있던지 간에 손쉽게 책의 Bar-Code 또는 QR-Code를 인식하여, 도서 정보를 제공 받을 수 있으며, 자신의 책을 관리 할 수 있는 유용한 기능을 제공한다.

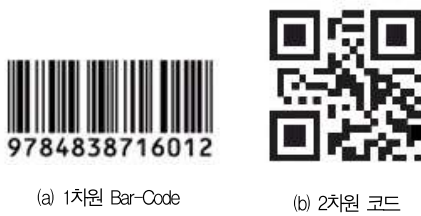
본 논문은 다음과 같이 구성된다. 2장에서는 어플리케이션에 사용되는 Bar-Code와 QR-Code에 대하여 소개하고, 코드 인식 라이브러리인 Zxing과 ZBar에 대하여 설명한다. 3장에서는 제안 어플리케이션의 구성과 전체적인 시스템 구조에 대하여 설명하며, 4장에서는 3장의 설계 내용을 바탕으로 개발된 어플리케이션의 실행 결과물을 제시한다. 그리고 5장에서는 도서 관리를 위한 데이터 처리 속도에 대한 비교 실험과 유사 어플리케이션의 결과 값에 대한 비교 실험 및 분석을 하며, 6장에서 결론을 맺는다.

II. 관련 연구

1.Bar-Code와 QR-Code

Bar-Code와 QR-Code는 대량의 데이터를 신속하고 정확하게 처리하기 위해 사용하는 정보 관리 방법이다. Bar-Code는 1923년 미국 윌리스 플린트가 '슈퍼마켓의 계산 자동화'에 대한 연구를 시작으로, 1차원 코드인 Bar-Code는 현재 슈퍼마켓 뿐 아니라, 물품의 생산과 유통, 사무실, 병원, 우체국, 통신, 항공 분야 등 전 산업계에서 널리 쓰이고 있다.

1차원 Bar-Code는 그림 1(a)와 같이 다양한 폭을 가진 검은 막대(Bar)와 흰 막대(Space)의 배열 패턴으로 정보를 표시한다. 그러나 데이터의 정확성, 심볼 크기, 정보량, 표현 정보의 단점이 있어 이를 해소하기 위해 2차원 코드가 만들어졌으며, 2차원 코드는 그림 1(b)와 같이 일정한 사격형의 크기에 X, Y 축으로 데이터를 2차원 심벌로 코드화 한 것이다.



(a) 1차원 Bar-Code (b) 2차원 코드
 그림 1. 1차원 Bar-Code와 2차원 코드
 Fig. 1. one and two-dimension bar code : (a) one-dimension bar code and (b) two-dimension code

2차원 코드는 데이터를 구성하는 방법에 따라 크게 다층형 바코드(Stacked Bar Code)와 매트릭스형 코드(Matrix Code)로 나뉜다. 대표적인 다층형 바코드는 PDF417라 불리는 Bar 코드으로써 1차원 Bar-Code와 같이 개별적으로 인식될 수 있는 몇 개의 문자가 모여 수평 방향 열을 구성한다[9]. QR 코드는 대표적인 매트릭스형 코드로써 정사각형의 동일한 폭의 흑백 요소를 모자이크 형식으로 배열하여 데이터를 구성한다[10]. QR 코드는 일본의 Denso Wave에 의해서 개발된 2차원 구조의 기호이며, 숫자의 경우 최대 4,296자, 한자, 한글의 경우 최대 1,817자까지 데이터를 담아 표현할 수 있다. 이외에도 DATA 코드, MAXI 코드 등 여러 종류의 2차원 Bar 코드가 존재하며, 2차원 Bar 코드는 대용량 데이터의 기록이 가능하여 숫자 또는 다양한 언어와 그래픽 데이터를 담을 수 있다.

2. Zxing과 ZBar

Zxing은 zebra crossing으로 불리며, 다양한 형태의 1D/2D Bar-Code를 인식할 수 있는 Java 기반의 오픈 소스이다. Zxing의 주요 목표는 서버와 통신하지 않고, 카메라가 있는 모바일 폰 자체에서 사진을 분석하고, 바코드를 디코딩하는 것이다. 현재 지원되는 코드 형태는 UPC-A and UPC-E, EAN-8 and EAN-13, Code 39, Code 93, Code 128, QR Code, ITF, Codabar, RSS-14, Data Matrix, PDF 417, Aztec 등이 있으며, Java 언어를 기반으로 만들어진 것이기 때문에 안드로이드 모바일 OS에는 위의 모든 코드를 다 인식할 수 있도록 구현되어 있다. 그리고 Java 언어를 C/C++언어로 변환하여, 아이폰 OS인 iOS에서도 사용 가능하게 하였으나, 현재 지원되는 코드는 QR-Code만 인식이 가능하다. 게다가 현재의 화면을 찍은 후 사진을 분석하는 형태를 취하고 있어, 인식이 안 될 경우 여러 번 찍어야 하는 불편함이 있다. 오픈 소스의 지원 형태는 예제 파일 형태로 지원되며, 다른 어플리케이션에서 적용하기 위해서는 필요 파일들을 복사하여 자신의 목적에 맞게 수정하

여 사용할 수 있다.

ZBar 또한 다양한 소스로부터 Bar-Code를 인식할 수 있는 오픈 소스이다. Zxing과 달리 C++을 기반으로 하였으며, EAN-13, UPC-A, UPC-E, EAN-8, Code 128, Code 39, 2/5 Code, QR-Code 등을 인식할 수 있다. ZBar는 C++을 기반으로 하였기 때문에 Linux/Unix, Windows, iPhone 등에서는 사용이 가능하나, 안드로이드 모바일 OS에서는 동작하지 않는다. 그러나 모바일 카메라로부터 실시간 스캐닝을 하여 사진을 분석하기 때문에 사용자가 수동으로 사진을 찍을 필요 없이 자동으로 Bar-Code를 트랙킹하고 인식하는 장점이 있다. 따라서 본 논문에서는 iOS에 보다 최적화 되어 있는 ZBar를 이용하여 Bar-Code와 QR-Code를 인식하게 하였다. ZBar의 오픈 소스 지원 형태는 .a 확장자를 갖는 라이브러리 파일이며, Zxing에 비해 코드가 복잡하지 않은 것이 특징이다.

III. 어플리케이션의 설계 및 구성

본 장에서는 제안 어플리케이션의 전체적인 시스템 구조를 설명하며, 어플리케이션의 동작 방법들에 대하여 설명한다. 제안 어플리케이션은 그림 2와 같은 순서로 도서 검색이 이루어진다.

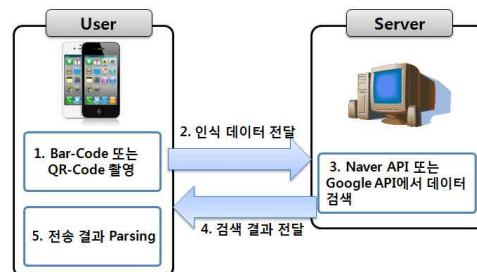


그림 2 도서 검색 어플리케이션의 동작
 Fig. 2. Process of book search application

책의 정보를 저장하려는 사용자는 아이폰의 카메라를 이용하여 책의 Bar-Code 또는 QR-Code를 촬영하며, 이때 자동으로 인식된 데이터는 보다 자세한 정보를 전달 받기 위해 Naver API 또는 Google API에 도서 검색을 요청한다. Naver API와 Google API를 동시에 사용하는 이유는 한국 도서의 경우 대부분 Naver API만으로 검색이 되나, 해외 도서의 경우 일부 검색되지 않는 도서를 Google API를 이용하여 보충하기 위함이다. 요청 방법은 Naver API와 Google API가 각각 다르기 때문에 표 1과 같이 구별되게 검색한다.

표 1. Naver와 Google API를 이용한 도서 검색 예
Table 1. Book search example using Naver and Google API

Naver	http://openapi.naver.com/search?key=user_key&query=9780135705995&target=book&d_isbn=9780135705995
Google	https://www.googleapis.com/books/v1/volumes?q=9780135705995+isbn:9780135705995&key=user_key

Naver와 Google API 모두 key 값을 전달하는데 이 값은 사용자에게 할당되는 값이며, 키 값에 해당하는 질의의 요청 횟수를 센다. Naver는 하나의 키에 대해 하루 최대 25,000 번 요청을 할 수 있으며, Google의 경우 실험실(Lab)로 운영되고 있어 검색 요청 횟수에 제한이 없다. 그리고 Bar-Code 또는 QR-Code로 인식된 데이터를 검색어로 전달해야 하기 때문에 Naver에서는 query 다음에 검색하고자 하는 질의 값을 입력하며, target에는 도서에 대해 검색하는 것으로 지정한다. 그리고 상세 질의의 경우 도서 ISBN으로 검색시 d_isbn이라는 추가 질의문을 함께 작성해야 결과를 얻을 수 있다. Google에서는 그보다 짧은 형태로 질의어는 q 다음에 작성하며, 추가 검색어로 isbn이라는 전달인자를 더 사용하여 검색을 진행한다. 즉, 표 1에서 요청한 질의는 Naver와 Google 모두 ISBN이 978-01-3570-599-5인 도서이며, 그 검색 결과 Naver에서는 XML(Extensible Markup Language) 형태인 그림 3과 같은 결과를 전송하며, Google에서는 JSON(JavaScript Object Notation) 형태인 그림 4와 같은 결과를 전송한다.

```
<?xml version="2.0"?>
<channel>
  <title>Naver Open API - book :: '9780135705995'</title>
  <link>http://search.naver.com/</link>
  <description>Naver Search Result</description>
  <lastBuildDate>Tue, 16 Aug 2011 19:34:08 +0900</lastBuildDate>
  <total>1</total>
  <start>1</start>
  <display>1</display>
  <item>
    <title>
      Computer Vision and Fuzzy-Neural Systems with CDROM
    </title>
    <link>
      http://openapi.naver.com/17
      AAc2LwQoCIRQAv0aPoj5dfQcPQexvhKbispUmFvj3bbAwh2Fg3p/Up0NjuQ8tuEbuDHy
    </link>
    <image>
      http://bookthumb.phinf.naver.net/cover/003/173/00317393.jpg?type=ml
    </image>
    <author>Kulkarni|Arun D.</author>
    <price>213103</price>
    <discount/>
    <publisher>PrenticeHallPTR</publisher>
    <pubdate>20010501</pubdate>
    <isbn>0135705991 <strong>9780135705995</strong></isbn>
    <description/>
  </item>
</channel>
</rss>
```

그림 3. Naver API를 이용한 도서 검색 결과 예시
Fig. 3. Book search result example using Naver API

```
{
  "kind": "books#volumes",
  "items": [
    {
      "kind": "books#volume",
      "id": "EeNRAAAAMAAJ",
      "etag": "p1GKfk0myV4",
      "selfLink": "https://www.googleapis.com/books/v1/volumes/EeNRAAAAMAAJ",
      "volumeInfo": {
        "title": "Computer vision and fuzzy-neural systems",
        "authors": [
          "Arun D. Kulkarni"
        ],
        "publisher": "Prentice Hall",
        "publishedDate": "2001-05-08",
        "industryIdentifiers": [
          {
            "type": "ISBN_10",
            "identifier": "0135705991"
          },
          {
            "type": "ISBN_13",
            "identifier": "9780135705995"
          }
        ]
      },
      "pageCount": 509,
      "printType": "BOOK",
      "categories": [
        "Computers"
      ],
      "contentVersion": "preview-1.0.0",
      "imageLinks": {
        "smallThumbnail": "http://bks5.books.google.com/books?id\u003dEeNRAAAAMAAJ&printsec\u003dfrontcover&img\u003d&zoom\u003d5&source\u003dgbgs_api",
        "thumbnail": "http://bks5.books.google.com/books?id\u003dEeNRAAAAMAAJ&printsec\u003dfrontcover&img\u003d&zoom\u003d1&source\u003dgbgs_api"
      },
      "language": "en",
      "previewLink": "http://books.google.com/books?id\u003dEeNRAAAAMAAJ&q\u003d9780135705995+isbn:9780135705995&dq\u003d9780135705995+isbn:9780135705995&ie\u003dISO-8859-1&cd\u003d1&source\u003dgbgs_api",
      "infoLink": "http://books.google.com/books?id\u003dEeNRAAAAMAAJ&q\u003d9780135705995+isbn:9780135705995&ie\u003dISO-8859-1&source\u003dgbgs_api"
    }
  ]
}
```

그림 4. Google API를 이용한 도서 검색 결과 예시
Fig. 4. Book search result example using Google API

그림 3의 XML 데이터 구성을 살펴보면, <title>이 책 제목을 나타내며, <link>는 상세 정보가 있는 주소를 나타낸다. 그리고 <image>는 책 표지 이미지 주소, <author>는 저자, <price>는 책 가격, <publisher>는 출판사, <pubdate>는 출판 날짜, <isbn>은 책 ISBN 숫자를 나타내며, <description>이 <description/> 으로 표시된 것으로 현재 Naver API에는 책에 대한 상세 설명이 없음을 볼 수 있다. 그림 4의 JSON 데이터 구성을 살펴보면, "volumeInfo" 안에 "title"이 책 제목을 나타내며, 그 아래에 "infoLink"는 상세한 정보가 있는 주소를 나타낸다. 그리고 "smallThumbnail"과 "thumbnail"이 크고 작은 책 표지 이미지 주소, "authors"에 저자, "publisher"에 출판사, "publishedDate"에 출판 날짜, "type": "ISBN_13"아래의 "identifier"에 책 ISBN을 나타낸 것을 볼 수 있다. 그리고 Google API는 상세 설명 데이터를 얻기 위해 "selfLink"의 주소를 한 번 더 요청하며, 그림 5와 같이 "description"에 상세 설명을 가져 온다.

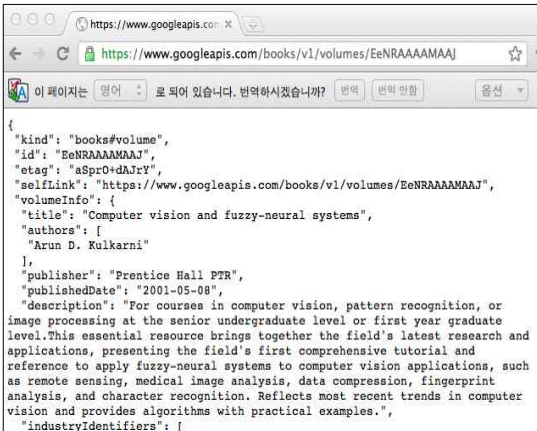


그림 5. Google API를 이용한 도서 상세 검색 결과 예시
Fig. 5. Book detail search result example using Google API

검색 요청에 의해 얻어진 결과는 다시 아이폰으로 전송이 되며, 전송 받은 데이터 처리를 위해 Naver API는 NSXML Parser를 사용하고, Google API는 JSON Parser를 사용한다. 각각의 Parser를 사용한 데이터는 책 표지와 함께 책 제목, 저자, 출판사, 출판 날짜, 상세 웹 사이트 링크 등을 결과 화면에 나타내며, 사용자는 그 결과를 스캔 목록(Scan List)에 저장하거나, 내 목록(My List)에 저장하여 도서를 관리 한다. 이때, 내 목록의 경우 저자별 목록을 보여주는 것이 필요하므로, 저자 목록을 위한 데이터 처리가 요구된다. 따라서 책 정보와 저자 정보를 위한 데이터 구성은 그림 6과 같이 한다.

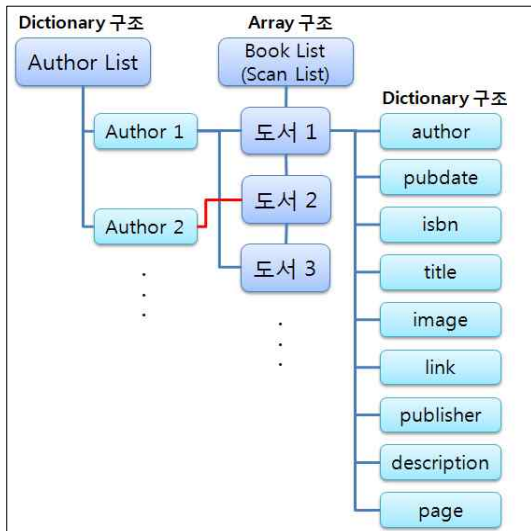


그림 6. 도서 검색 결과의 데이터 처리 구조
Fig. 6. Data process structure of book search result

그림 6에서 Book List와 Scan List은 스캔 목록과 내 목록을 나타낼 때 사용하며, 그 세부 내용은 각 도서 오른쪽에 있는 Dictionary 구조의 데이터를 표현한다. 그리고 저자별 목록을 보여주기 위해서는 Author List에 저자에 연결된 각각의 도서들을 불러와 사용하게 된다. 이때 도서의 아래에 있는 page 노드는 초기 데이터는 0으로 가지며, 사용자가 차후 자신이 읽은 페이지 수를 입력하면 그때 이 노드를 사용하게 된다.

IV. 어플리케이션의 개발 결과

본 장에서는 3장에서 설계한 내용을 바탕으로 개발된 어플리케이션의 결과물을 보여준다. 그림 7은 제안 어플리케이션의 첫 화면이다. 사용자는 도서 정보를 입력하기 위해 Book Scan 메뉴를 선택하여 책의 Bar-Code 또는 QR-Code를 스캔한다. Book Scan 메뉴가 선택되면 어플리케이션은 아이폰의 ImagePicker를 이용해 내장된 카메라를 띄우며, ZBar 라이브러리가 동작하여 실시간으로 코드를 스캔한다. 그림 8은 “Computer Vision and Fuzzy-Neural Systems”이라는 책을 스캔한 결과 화면이다. Naver API에서는 책 요약 결과를 얻을 수 없었기 때문에, 어플리케이션은 Google API에서 얻어온 결과로 표현하였으며, 그림 8의 가운데 오른쪽 “web” 글씨를 터치하면 그림 9와 같이 상세 정보 웹 사이트 또한 Google로 연결되는 것을 볼 수 있다.

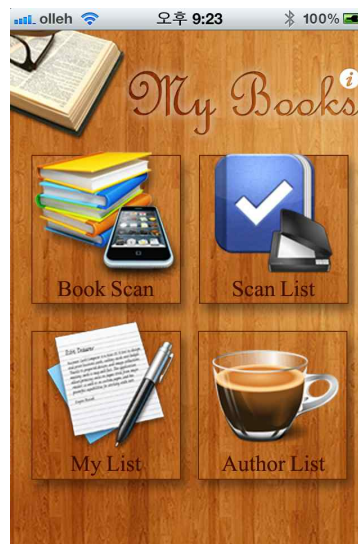


그림 7. 도서 관리 어플리케이션의 첫 화면
Fig. 7. Main scene of book management application

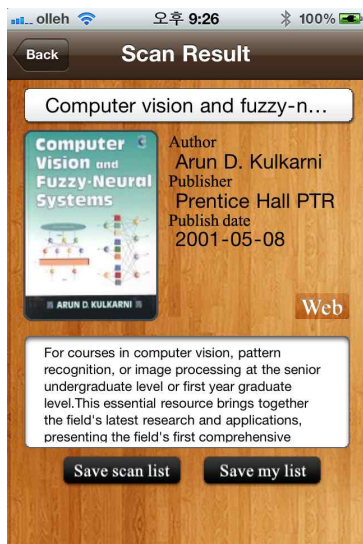


그림 8. 도서 관리 어플리케이션의 검색 결과 화면
Fig. 8. Search result scene of book management application

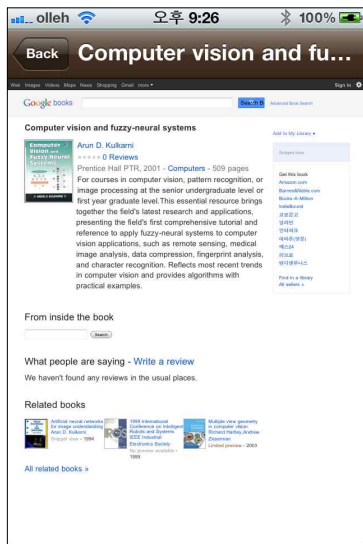


그림 9. 검색 결과 상세 보기 중 웹 뷰 화면
Fig. 9. Web view scene of search result detail view

사용자는 검색된 결과 내용을 스캔 목록 또는 내 목록으로 저장할 수 있다. 스캔 목록은 자신이 스캔한 책들만 저장하는 것으로 구입하지 않았거나, 도서관, 서점 등에서 잠시 보기 위한 용도로 만든 목록이다. 내 목록은 자신이 구입한 책을 정리하고, 그 책에 대하여 소감, 코멘트 등을 작성하는 용도

로 만든 목록이다. 그림 10은 내 목록에 저장한 도서의 목록을 나타내는 화면으로 그림과 같이 내 목록, 스캔 목록 모두 최근 등록된 도서가 맨 위에 나타나게 된다. 사용자는 등록된 책을 삭제하거나, 등록 순서를 바꾸고 싶은 경우 그림의 오른쪽 위에 있는 Edit를 터치하여 삭제, 순서 변경을 할 수 있다. 그림 11은 내 목록에 저장된 책을 보는 화면으로 사용자가 남긴 코멘트가 함께 보인다.



그림 10. 내 목록 보기 화면
Fig. 10. My list view scene

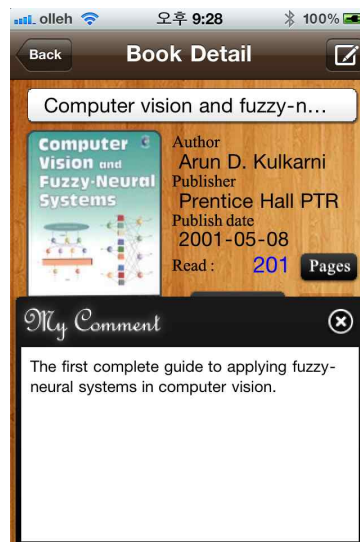


그림 11. 내 목록에 있는 책의 상세 보기 화면
Fig. 11. Detail view scene of book in my list

그림 11의 오른쪽 위에 글쓰기 같은 아이콘이 코멘트를 작성하는 메뉴이며, 스캔 목록에서 책 상세 보기를 하는 경우 나타나지 않는 메뉴이다. 그리고 그림 중앙의 오른쪽 편에 Pages를 터치하면, 자신이 현재까지 읽은 책 Page를 기록할 수 있으며, 그 기록 내용이 옆에 숫자로 표시 되는 것을 볼 수 있다. 그리고 내 목록에 저장한 도서는 저자별로 볼 수 있는 목록을 제공한다. 그림 7의 Author List가 이에 해당하는 메뉴이며, 사용자가 터치 시 그림 12와 같은 목록과 인덱스를 볼 수 있다.

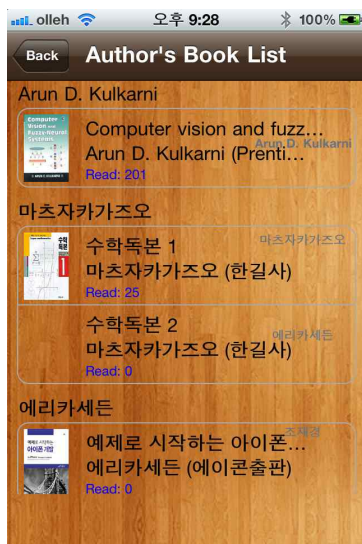


그림 12 저자별 내 목록 보기 화면
Fig. 12. My list view scene according to author group

지금까지의 어플리케이션 설명에서와 같이, 제안 어플리케이션은 자신의 도서를 정리하고, 자신이 읽던 책이 어디까지 읽었는지 등을 저장하는데 있어 유용함을 확인 할 수 있다. 그리고 기존 Apple사에서 제공하는 Books 어플리케이션의 기능을 그대로 작은 모바일기기에 담은 것임을 볼 수 있다. 따라서 책이 많아 도서 정리가 힘든 사용자, 도서관 혹은 서점에서 잠시 잠깐 책을 보는 사용자, 사고 싶은 책 이름을 저장해 놓고 싶은 사용자들에게 제안 어플리케이션은 유용하게 쓰일 것이다.

V. 실험 및 분석

본 장에서는 제안 어플리케이션의 유용성을 검증하기 위해

어플리케이션의 데이터 처리 속도에 대한 비교 실험과 유사 어플리케이션의 검색 결과에 대한 비교 실험을 하고, 이를 분석한다.

1. 데이터 처리 속도 실험

데이터 처리 속도 실험을 위해 대학생 남녀 구분 없이 20명을 선정 하였으며, 각 인원 당 도서 1권에 대한 입력 속도를 측정했다. 입력 방법은 제안 어플리케이션을 이용하지 않고, 스마트폰 내의 카메라를 이용한 촬영 후, 책 제목을 입력하는 A 그룹(10명)과 제안 어플리케이션을 이용하는 B 그룹(10명)으로 나누었다. 이때, B 그룹에서 데이터 정보를 받아 오기 위한 네트워크 상태는 WIFI 연결이 아닌 3G를 사용하였다. 표 2는 처리 속도 실험에 대한 평균 시간을 나타낸 것이다.

표 2 도서 데이터의 처리 속도 비교
Table 2. Processing speed comparison of book data

동작	A 그룹	B 그룹
사진 촬영/ 바코드 인식	4.8 초	2.4초
제목 입력	8.9 초	4.5 초
합계	13.7 초	6.9 초

표 2에서와 같이 A 그룹의 경우 사진 촬영을 하는데 평균 4.8초의 시간이 걸린 반면, B 그룹은 바코드 인식을 위해 2.4초로 처리 시간이 1/2로 줄어든 것을 볼 수 있다. 이는 A 그룹의 경우 도서의 사진을 찍기 위해 촬영하는 사람이 도서의 각도, 위치 등을 고려하고, 촬영하기 위한 도서의 초점을 맞추는데 시간이 오래 걸렸기 때문임을 예상할 수 있다. 반면 B 그룹은 도서의 뒷면에 바코드만 인식하기 위해 초점이나 도서의 각도, 위치 등을 고려하지 않고 어플리케이션 스스로 바코드를 인식하는 시간만을 기다렸기 때문에 처리 속도가 더 빠른 것이다. 제목 입력에서도 A 그룹은 평균 8.9초인 반면, B 그룹은 4.5초가 걸리는 것을 확인 할 수 있다. A 그룹은 도서 제목을 일일이 텍스트로 입력해야 하며, 입력 중 오타가 나서 지우고 다시 입력하는 경우도 발생하기 때문에 오랜 시간이 걸리는 것이다. 반면 B 그룹은 바코드 인식 후 자동으로 3G 네트워크를 통해 제목이 입력되므로, 데이터를 요청하고

전송 받는 시간만 필요하기 때문에 처리 속도가 빠른 것이다. 따라서 전체적인 처리 속도를 비교할 때, 제안 어플리케이션이 수동으로 입력하는 것보다 빠름을 알 수 있다.

2. 유사 어플리케이션과의 검색 실험

제안 어플리케이션의 도서 검색 성능 테스트를 위해 국내 도서 20권과 해외 도서 20권을 선정하였으며, 비교를 위한 어플리케이션은 Bar-Code나 QR-Code를 인식하는 기존 상용화된 App인 QRDic과 쿠루쿠루를 이용하였다. 이때, 도서 선정 방법은 국내 도서의 경우 2011년 10월 1주, 인터넷 교보문고 종합 베스트셀러 순위 20위 리스트의 해당 도서를 선택하였으며, 국외 도서의 경우 아마존의 2011년 10월 4일 Best sellers in Books의 20위 리스트의 해당 도서를 선택했다. 표 3은 국내 도서와 국외 도서의 Bar-Code와 QR-Code를 인식하고, 그에 해당하는 도서를 결과로 정확히 출력하였는지 평균을 낸 것이다.

표 3. 도서 검색 결과의 정확도 비교
Table 3. Accuracy comparison of book search result

	제안 어플리케이션	QRDic	쿠루쿠루
국내 결과	100%	100%	100%
국외 결과	85%	25%	25%
합계	92.5%	62.5%	62.5%

표 3에서와 같이 국내 도서의 검색 결과는 모두 100% 보 여주는 것을 볼 수 있다. 그러나 국외 도서의 경우 제안 어플리케이션은 85%의 검색 결과를 보이는 반면, 다른 어플리케이션들은 모두 25%라는 낮은 검색 결과를 보였다. 이는 제안 어플리케이션의 경우 도서 정보를 Naver API를 이용하면서도 도서가 검색되지 않은 경우 Google API까지 검색하기 때문이다. 반면 QRDic과 쿠루쿠루는 국내의 API만을 사용하기 때문에 해외 도서의 검색 결과가 좋지 못한 것으로 보인다. 즉, 제안 어플리케이션은 국내, 국외 모든 도서에서 유용하게 사용 가능하다는 것을 볼 수 있었다.

VI. 결 론

본 논문에서 제안한 어플리케이션은 사용자가 아이폰의 카메라를 이용해 Bar-Code 또는 QR-Code를 인식하여 책의 상세 정보를 볼 수 있을 뿐 아니라, 자신이 가지고 있는 도서를 정리, 읽은 소감, 코멘트 등을 작성할 수 있다. 그리고 도서관이나 서점에서도 간편히 책을 스캔하여 간단한 소개 글을 볼 수 있으며, 자신이 어디까지 읽었는지를 저장할 수 있다. 기존 네트워크가 되는 고가의 컴퓨터, 카메라를 사용해서 해야 하는 부분들을 고스란히 모바일로 옮겼을 뿐 아니라 간편함을 추가하여 기존 기술을 발전시킨 것이라 할 수 있다. 차 후 연구로는 같은 책을 소장하고 있는 사람간의 소통을 할 수 있는 서버를 마련하여 SNS (Social Networking Service) 기능을 추가한 서비스를 개발할 수 있으며, 동일 도서를 갖은 사람들 간에 추천 서비스를 할 수 있는 알고리즘 연구를 할 수 있을 것이다.

참고문헌

- [1] P. JungHyun, "Mobile Marketing of Smart Phone Period," Marketing, Vol. 45, No. 1, pp. 33-38, January 2011.
- [2] O. Oda, and S. Feiner, "Rolling and shooting: two augmented reality games," 28th International Conference Extended Abstracts on Human in Computing Systems, pp. 3041-3044, April 2010.
- [3] C. T. Tan, and D. Soh, "Augmented Reality Games: A Review," GAMEON-ARABIA, EUROSIS, October 2010.
- [4] iPhone ARKit Project, <http://code.google.com/p/iphonearkit>
- [5] R. Muniz, L. Junco, and A. Otero, "A Robust Software Barcode Reader Using the Hough Transform," International Conference on Information Intelligence and Systems, pp. 313-319, October 1999.
- [6] E. Ohbuchi, H. Hanaizumi, and L.A. Hock, "Barcode readers using the camera device in mobile phones," International Conference on Cyberworlds, pp. 260-265, November 2004.
- [7] Multi-format 1D/2D barcode image processing library, <http://code.google.com/p/zxing/>
- [8] ZBar bar code reader, <http://zbar.sourceforge.net/>

- [9] ISO/IEC 15420:2000, "Automatic identification and data capture techniques. Bar code symbology specification. EAN/UPC," Information technology, 2000.
- [10] ISO/IEC 18004:2000, "Automatic identification and data capture techniques. Bar code symbology. QR code," Information technology, 2000.

저 자 소 개



정 명 범

2004 : 송실대학교 미디어학부
공학사.

2006 : 송실대학교 미디어학과
공학석사.

2010 : 송실대학교 미디어학과
공학박사

현 재 : 송실대학교 미디어학과
박사후 과정

관심분야 : 콘텐츠 공학, 모바일 콘
텐츠

Email : nzin@ssu.ac.kr



강 미 애

1997 : 송실대학교 정보과학대학원
공학석사.

현 재 : 송실대학교 IT정책경영학과
박사과정 (주)야긴스텍

관심분야 : 위치정보서비스, 모바일
보안

Email : makang@yagins.com



고 일 주

1992 : 송실대학교 전산학과 공학사.

1994 : 송실대학교 전산학과 공학
석사.

1997 : 송실대학교 전산학과 공학
박사

현 재 : 송실대학교 미디어학부
부교수

관심분야 : 콘텐츠, 영상 처리, 감
성 공학

Email : andy@korea.ac.kr