

도면 출력 안드로이드 애플리케이션 구현

임재걸*, 윤승호*, 진창훈*

*동국대학교 컴퓨터멀티미디어학과

e-mail : {yim, ysh0605, jowk80}@dongguk.ac.kr

Implementation of an Android Application of Rendering Drawings

Jaegel Yim*, Seung-Ho Yoon*, Chang-Hoon Jin*

*Dept of Computer and Multimedia, Dongguk University

요 약

자동차 내비게이션을 비롯한 각종 위치기반서비스가 일상생활에 큰 도움을 준다는 단언에 대하여 반론을 제기하는 사람은 없을 것이다. 고층 건물과 지하 도시가 개발됨에 따라 사람들이 옥내에서 생활하는 시간이 늘어나게 되고 따라서 미구에 옥내위치기반 서비스에 대한 요구가 옥외서비스에 대한 요구보다 더 늘어날 것으로 예상된다. 또한 스마트 폰이 널리 보급되어 있으므로 옥내위치기반서비스가 제공되는 플랫폼으로는 스마트 폰이 제격이다. 옥내위치기반서비스의 사용자 인터페이스에는 당연히 도면이 보여야한다. 이에 본 논문은 도면 출력 안드로이드 애플리케이션을 개발한다.

키워드: 안드로이드; 도면; 옥내위치기반서비스; 도면출력; 애플리케이션

1. 서론

인구가 과밀하게 됨에 따라 거대한 건물과 복잡한 지하상가가 점점 더 많아지게 되고, 더욱 많은 사람들이 더욱 오랜 시간을 옥내에서 생활하게 되었다. 이에 따라 옥내 위치기반서비스에 대한 요구도 점점 더 절실해지고 있다. 한편 안드로이드는 휴대폰용 운용체제, 미들웨어, 응용프로그램을 한데 묶은 소프트웨어 플랫폼으로서 공개 소프트웨어이기 때문에 최근 다수의 스마트 폰이 안드로이드를 채용하고 있다. 따라서 안드로이드 폰을 위한 옥내위치기반서비스(ILBS: Indoor Location Based Service) 애플리케이션의 효용성은 지대할 것이다.

무릇 위치기반서비스의 사용자 인터페이스에서 가장 중요한 요소는 지도 혹은 옥내의 경우에는 도면이다. 도면에 사용자의 현재 위치를 표시해주고 경로 혹은 유명지형지물을 표시해주어야 사용자가 상황을 파악할 수 있다. 따라서 도면을 출력하여주는 프로그램은 이미 무수히 개발되었다. 그러나 안드로이드 스마트폰용 도면 출력 애플리케이션은 아직 개발된 바가 없다.

한편 대부분의 건물 도면은 오토캐드로 작성되며, 모든 건물에 대한 도면은 반드시 존재함으로, 본 논문은 오토캐드 도면을 출력하는 안드로이드 애플리케이션을 개발한다. 본 애플리케이션은 오토캐드 도면 중 DXF 도면 파일을 입력으로 받아 스마트폰에 출력하고, 이동과 확대 및 축소 기능도 제공한다.

2. 관련연구

도면 출력 및 도면 사용 관련 연구는 지리학과 건축 분야에서 널리 수행되었다. [1]은 건물등록을 위하여 ArcGIS 엔진을 이용하여 개발한 자동등록 방법을 적용하는 실험 결과의 오차를 분석하고, 실용적이며 효율성을 획기적으로 제고함을 보였다. [2]는 건설공사 준공도면을 출력하고, 이를 활용하여 수치지도를 수시로 갱신할 수 있는 방안을 제시하였다. 설계 과정의 효율을 제고하기 위하여 건축 설계 분야에서는 BIM(Building Information Modeling)이 주요 쟁점으로 떠오르고 있다. [3]은 2D 도면의 표준화가 BIM 표준에 매우 중요한 위치를 차지하고 있음을 밝히고, BIM이 한국 건설 분야에 도입되는데 일조하기 위하여 전자 2D 도면 정보의 표준을 개선한다. [4]는 시설물 도면을 실무자가 작업현장에서 PDA에 출력하여 봄으로써 시설물 현황을 실시간으로 직접 확인할 수 있고, 작업현장에서 실시간으로 GIS 데이터베이스를 구축할 수 있는 시스템의 실용성을 평가한 결과를 소개한다. 본 평가를 위하여 PDA에 도면을 보여주는 파일럿 시스템을 개발하였다.

위에서 살펴본 바와 같이 도면 출력은 건물등록, 수치지도 갱신, 건축설계, GIS 데이터베이스 구축 등 다양한 작업을 수행하는 시스템에 이용된다. 본 연구가 개발한 도면 출력 안드로이드 애플리케이션은 옥내위치기반서비스의 인터페이스가 주목적이나 상기 응용들을 스마트폰에 제공하는 징검다리 역할도 할 것이다.

3. DXF 도면 파일 형식

이 논문은 2011년도 정부(중소기업청)의 지원을 받아 수행된 산학연 공동기술개발사업임(R00046281)

오토캐드(AutoCAD)는 2차원/3차원 디자인, 제도를 위한 CAD 응용 소프트웨어이다. 오토캐드의 파일 형식은 기본 형태로 DWG이지만, 다른 소프트웨어와 상호연동을 위하여 DXF도 지원한다. 즉, DXF 파일 형식은 다른 소프트웨어에서 읽기에 편리하다는 특징이 있다. 예를 들어, Autocad로 작성한 도면이나 지도를 DXF 형식으로 저장하면 일반적인 텍스트 파일로 저장됨으로 메모장이나 다른 소프트웨어로 파일 내용을 읽어 볼 수 있다.

DXF 파일 형식과 관련하여 본 논문에서 필요로 하는 사항들은 다음과 같다. DXF 파일에서는 정보가 그룹코드와 쌍을 이루어 저장된다. 이때 그룹코드는 정보의 유형을 지정하는 역할을 한다. 정보의 유형에는 텍스트, 정수, x좌표, 등 여러 가지가 있다. 예를 들어 표 1에서 그룹코드 0은 어떤 정보 단위의 시작이나 끝임을 나타내며 쌍을 이루는 항은 어떤 유형의 정보단위인지를 나타낸다. 즉 표 1의 두 번째 줄의 경우에는 LINE이라는 정보 단위의 시작임을 나타낸다. LINE 정보를 나타내는 영역에서, 그룹코드 10은 쌍을 이루는 항이 선의 시작점의 x좌표, 20은 y좌표, 30은 z좌표임을 의미하고, 그룹코드 11은 쌍을 이루는 항이 선의 끝점의 x좌표, 21은 y좌표, 31은 z좌표임을 의미한다.

반지름의 길이, 호의 시작 각 그리고 끝 각으로 구성된다.

4. 설계

본 연구에서 구현하는 지도보기 프로그램은 Autocad로 작성한 DXF 유형의 지도 파일을 입력받아 화면에 출력하고 확대와 축소 등의 조작 기능을 제공한다. 이러한 지도보기 모듈은 입력부분, 그리기부분, 제어부분으로 구성된다.

4.1 입력부분

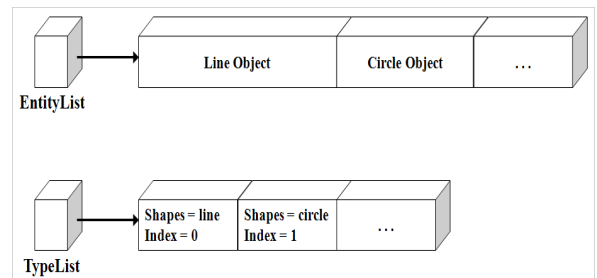
입력 부분은 DXF 파일을 도형의 유형(LINE, CIRCLE, ARC, ...) 단위로 읽어 해당 개체를 생성하여 엔터티 리스트에 추가한다. 이때 엔터티 리스트에 추가된 개체의 유형 이름을 인덱스와 함께 유형 리스트에 추가하여 둔다. 이렇게 하는 이유는 엔터티 즉 도형의 종류가 여러 가지인데 모든 유형의 엔터티들을 하나의 리스트에 모두 기록함으로써 새로운 유형의 엔터티가 추가로 만들어지더라도 본 소프트웨어를 그대로 사용할 수 있도록 하기 위함이다. 예를 들어 표 1의 DXF 파일을 입력하면 엔터티리스트(EntityList)와 유형리스트(TypeList)는 <그림 1>과 같이 작성된다.

<표 1> DXF 파일의 예

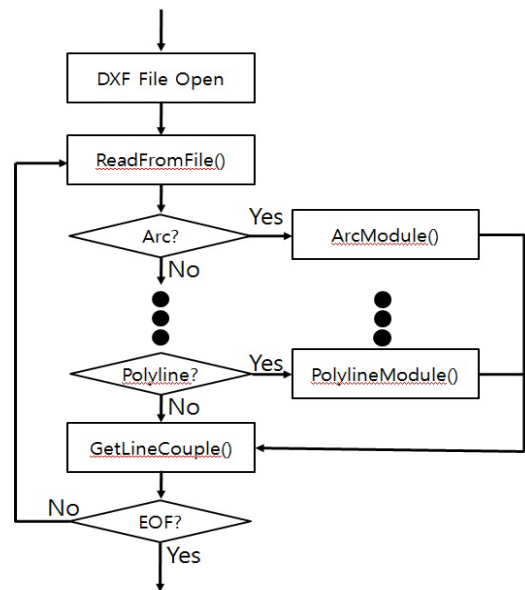
Table 1. An example DXF File

...	...
0	엔터티의 종류
LINE	line
8	이 엔터티가 속한 레이어명은 "0"
0	
10	라인의 시작
35.88862818985399	x 좌표 값
20	라인의 시작
13.65995672501049	y좌표 값
30	라인의 시작
0.0	z좌표 값
11	라인의 끝
58.15630849450378	x좌표 값
21	라인의 끝
40.68115047870072	y좌표 값
31	라인의 끝
0.0	z좌표 값
0	엔터티 종류
CIRCLE	circle
8	이 엔터티가 속한 레이어명은 "0"
0	
10	중심점
46.20050706958136	x 좌표
20	중심점
32.17171374022175	y 좌표
40	반지름의
6.2544244956812	길이
...	...

엔터티의 종류에 따라 엔터티 정보를 구성하는 요소가 각각 다르다. LINE의 경우에는 시작점의 좌표와 끝점의 좌표로 엔터티 정보가 구성되며, 원의 경우에는 중심점의 좌표와 반지름으로 구성된다. 한편 호는 중심점의 좌표,



<그림 1> 표 1의 입력 결과



<그림 2> DXF 파일을 읽어오는 과정의 순서도

DXF 파일로부터 도면정보를 읽어오는 과정을 순서대로 보이면 <그림 2>와 같다. DXFImport Class에서 DXF 파일을 연 다음, ReadFromFile() 메소드가 호출되어 GetLineCouple() 메소드를 통해 DXF 파일에서 EOF까지 두 줄씩 데이터를 읽어온다. 표 1의 DXF 파일 예에서 알 수 있는 바와 같이 현재 읽은 줄이 새로운 엔터티의 시작인지 아닌지는 읽어 온 두 줄 중 첫줄이 "0"이고 다음 줄이 "LINE", "CIRCLE", 등의 도형 유형 이름인지 보면 알 수 있다. 따라서 현재 읽은 줄이 엔터티 정보의 시작이면 해당 엔터티 정보를 읽는 메소드를 호출한다.

엔터티 정보를 읽는 각 메소드는 해당 도형을 결정하는 요소들을 입력받아 엔터티 개체를 생성하여 엔터티 리스트에 추가한다. 그림 1에 보이는 바와 같이, 이때 엔터티 유형과 해당 개체의 엔터티 리스트에서의 위치를 나타내는 인덱스를 유형리스트에도 추가한다.

4.2 그리기부분

그리기 메소드는 유형리스트를 구성하는 개체들을 하나씩 차례로 접근하여, 이 개체의 인덱스 필드의 값으로 이 개체와 연관된 엔터티 리스트의 개체를 사용자 인터페이스인 윈도우에 그림으로써 전체 지도를 화면에 그린다. 각 엔터티 개체(예를 들면 Line 개체)에는 그리기 메소드가 멤버메소드로 정의되어 있으며, 도형을 결정하는 값들뿐만 아니라 지도의 중앙점 좌표 (paperMidX와 paperMidY), 윈도우의 중앙점 좌표 (windowX, windowY), 스케일, 등도 인수로 받는다. 각 인수들의 값을 구하는 방법은 다음과 같다.

지도의 중앙점 좌표는 출력될 지도의 중앙점으로, 지도 좌표들 중 X좌표의 최대값과 최소값의 차이의 반을 최소값에 더한 값이 지도 중앙점의 X좌표이며 중앙점의 Y좌표도 같은 방법으로 구한다. 일반적으로 지도의 중앙점을 구하는 식은 다음과 같다:

$$\text{paperMidX} = \text{XMin} + (\text{XMax} - \text{XMin}) / 2;$$

$$\text{paperMidY} = \text{YMin} + (\text{YMax} - \text{YMin}) / 2;$$

지도의 중앙점은 사용자가 지도를 상하좌우로 움직일 때 변하게 된다.

윈도우의 중앙점 좌표는 지도가 출력될 영역의 중앙점으로 다음 식으로 구한다.

$$\text{windowMidX} = \text{drawPanelWidth} / 2;$$

$$\text{windowMidY} = \text{drawPanelHeight} / 2;$$

지도 좌표를 윈도우 좌표로 변환하기 위하여 지도 영역의 크기에 대한 윈도우 영역의 크기의 비율, 즉 스케일을 구한다. x 스케일과 y 스케일 중, 작은 것을 적용해야 지도 전체를 윈도우에 나타낼 수 있다. 또한 이 스케일을 그대로 적용하면 지도가 화면에 꽉 들어참으로 90%로 줄인 스케일을 그리기에 적용한다.

그리기는 입력에서 작성한 엔터티 리스트를 구성하는 엔터티들을 하나씩 차례로 화면에 출력하는데, 이때 지도

좌표 (paperX, paperY)를 비롯한 엔터티를 결정하는 요소들을 지도의 중앙점 (paperMidX와 Y), 윈도우의 중앙점 (windowMidX와 Y) 그리고 스케일을 사용하여 윈도우 좌표 (winX, winY)로 변환한다. 좌표변환 방법은 주어진 지도 좌표와 지도의 중앙점 간의 거리에, 스케일을 곱한 결과를 화면의 중앙점 좌표에 더한다. 이 방법을 식으로 표현하면 다음과 같다. 이때 윈도우의 Y좌표는 지도의 Y좌표와 반대로 위에서 아래 방향으로 증가하기 때문에 음수를 취해준다.

$$\text{winX} = (\text{paperX} - \text{paperMidX}) * \text{mainScale} + \text{windowMidX};$$

$$\text{winY} = -(\text{paperY} - \text{paperMidY}) * \text{mainScale} + \text{windowMidY};$$

4.3 제어부분

화면에 출력된 지도를 조작하기 위한 사용자 인터페이스로 ZoomIn, ZoomOut, Up, Down, 등의 버튼을 제공한다. Button_ZoomIn() 메소드가 호출되면 현재 Scale을 증가(mainScale * 1.2)시켜 도면을 확대하며, Button_ZoomOut() 메소드가 호출되면 현재 Scale을 감소(mainScale / 1.2)시켜 도면을 축소한다. 이후 최초의 도면크기를 원할 때 RecalculateScale() 메소드를 호출하면 된다.

Button_Up(), Button_Down(), Button_Left(), Button_Right() 메소드는 도면을 상하좌우로 이동하는 메소드이며, PaperToWindow() 메소드는 도면의 좌표를 DrawPanel 좌표로 변환하고, WindowToPaper() 메소드는 DrawPanel의 좌표를 도면의 좌표로 변환해 주는 메소드이다. <표 2>에서는 각 메소드에 대해 계산하는 방법을 나타내고 있다.

<표 2> Control Part의 메소드

메소드	계산 방법
Button_ZoomIn()	mainScale = mainScale * 1.2;
Button_ZoomOut()	mainScale = mainScale / 1.2;
Button_Up()	paperMidY = paperMidY + 30 / mainScale;
Button_Down()	paperMidY = paperMidY - 30 / mainScale;
Button_Left()	paperMidX = paperMidX - 30 / mainScale;
Button_Right()	paperMidX = paperMidX + 30 / mainScale;
WindowToPaper()	paperX = (winX + paperMidX * mainScale - windowMidX) / mainScale; paperY = (-winY + paperMidY * mainScale + windowMidY) / mainScale;
PaperToWindow()	winX = (paperX - paperMidX) * mainScale + windowMidX; winY = -(paperY - paperMidY) * mainScale + windowMidY;

5. 구현

도면을 확대하거나 이동하면 도면이 화면 전체에 그려지는 현상이 나타난다. 이에 대한 해결 방안으로 Bitmap의 크기를 정의하여 그 위에 도면을 그린다. 아래는 Bitmap을 크기 450*500, 색상포맷은 config.ARG888로 각각 설정한다. Bitmap위에 도면을 그리기 위하여

Canvas를 설정한다. Canvas 설정을 하면 Bitmap위에 그림을 그릴 수 있고, Canvas 출력 시 Bitmap 위에 그려진 도면이 출력된다. 또한 도면의 가독성을 높이기 위해서 가로와 세로를 각각 15픽셀만큼 여백을 준다.

```
public void Draw(Canvas g)
{
    Bitmap bitmap =Bitmap.createBitmap(450, 500,
    Config.ARGB_8888);
    Canvas test = new Canvas(bitmap);
    .
    .
    .
}
g.drawBitmap(bitmap, 15, 15,null);
}
```

사용자의 편리성을 위해 버튼을 넣어서 화면의 확대, 축소, 이동을 하였다. 아래는 도면의 확대·축소에 대한 함수를 보인다. 이때 확대는 mainScale에 1.2를 곱하여 배율을 높여주고, 축소는 mainScale에 1.2를 나누어 배율을 낮춰줌으로써 각각의 기능을 구현하였다.

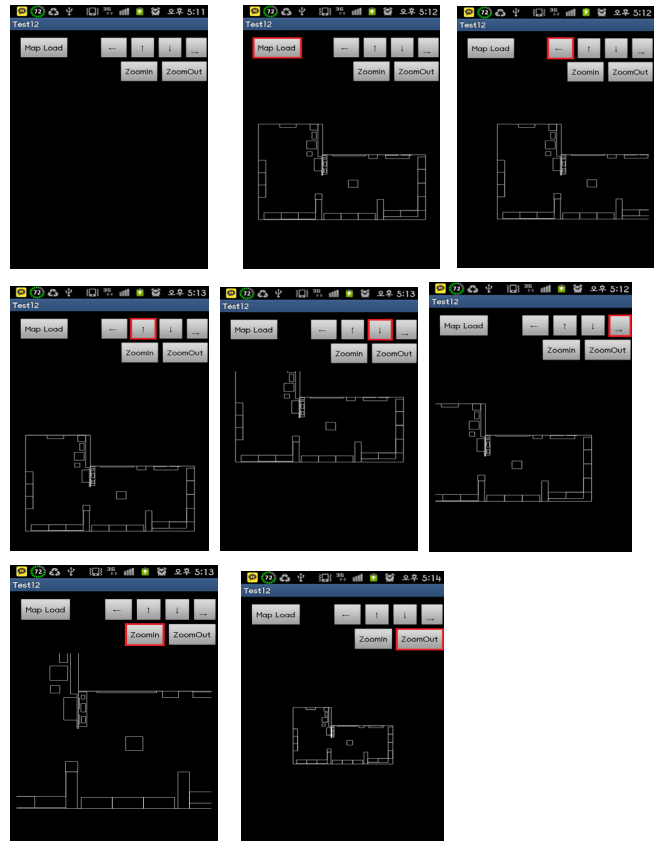
```
public void Button_ZoomIn()
{
    mainScale = mainScale * 1.2;
}
public void Button_ZoomOut()
{
    mainScale = mainScale / 1.2;
}
```

아래는 도면의 이동에 해당하는 상·하·좌·우 각각의 버튼 클릭에 대한 함수를 보인다. 이때 상·하 버튼은 도면의 Y좌표 중심인 paperMidY의 좌표에 30/mainScale을 가감해준다. 또한 좌·우 버튼은 도면의 X좌표 중심인 paperMidX의 좌표에 30/mainScale을 가감해준다. 이때 30은 개발자가 임의로 지정한 값이다.

```
public void Button_Up()
{
    paperMidY = paperMidY + (double)30 / mainScale;
}
...
public void Button_Right()
{
    paperMidX = paperMidX + (double)30 / mainScale;
}
```

6. 실험

도면 출력 애플리케이션을 갤럭시S2 기종에서 실험한 결과가 그림 3에 보인다.



<그림 3> 도면 출력 애플리케이션 실행 화면: 초기화면, 도면출력, 이동, 확대 및 축소 결과 화면이 보임.

6. 결론

위치기반서비스는 사용자에게 매우 유용한 정보를 제공함으로써 옥내·외 할 것 없이 제공되어야 한다. 특히 옥내위치기반서비스 제공 시에는 현재 위치를 알려줄 수 있는 전자 도면의 중요성이 크다. 본 논문에서는 오토캐드를 이용하여 전자 지도를 작성하고, DXF 파일 형식으로 변환하여 지도의 가용성을 높였으며, 또한 안드로이드를 이용하여 스마트폰 내에 전자 도면 출력 애플리케이션을 구현하였다.

참고문헌

[1] 박기현, 엄정섭, “건축도면을 활용한 지적도의 건물 등록 정확도 평가,” 한국GIS학회 2007 공동추계학술대회 논문집 2007.6, pp. 233-239
 [2] 신동빈, 유선철, "수치지도 갱신을 위한 건설공사 준공도면 활용 문제점에 관한 연구," 한국GIS학회 2008 공동추계학술대회 2008.6, pp. 19-25
 [3] 권오철, 조찬원, “BIM도입을 고려한 2D 전자도면 표준 발전방향에 관한 연구,” 대한건축학회 논문집-계획계, 제24권 제5호 (통권 235호) 2008.5, pp. 49-57
 [4] 정지훈, 엄정섭, “대학시설물 관리에서 PDA 기반의 실시간 Mobile GIS 도입 타당성 평가,” 한국GIS학회지, Vol.11, No.1, 2003년, pp. 41-60