

이종 모바일 장치에 호환성 있는 모바일 GIS 엔진 개발

홍승욱* · 박재양** · 박수현***

*동서대학교 디자인&IT 전문대학원, ** (주)유비텍, ***동서대학교 컴퓨터정보공학부

Development of compatible Mobile GIS Engine in heterogeneous mobile devices

Seung-wook Hong* · Jae-yang Park** · Su-hyun Park***

*Dongseo University Graduate School of Design&IT · **Ubitec · ***Dongseo University Department of Computer Engineering

E-mail : hswchaos@nate.com subak@gdsu.dongseo.ac.kr

요 약

최근 모바일 장치에서 GIS를 활용한 다양한 서비스 시스템들이 연구·개발되어지고 있다. 하지만 모바일 장치는 하드웨어 구성의 차이와 운영체제에 따른 제공 라이브러리의 차이가 심하다. 그래서 대부분의 서비스 시스템이 특정 모바일 장치에 제한적으로 설계·구현되어지고 있다.

본 연구에서는 기존 모바일 GIS엔진에서 디자인 요소와 같이 특정 모바일 장치에 제한적일 수밖에 없는 부분을 제외한 모바일 GIS엔진을 설계하였다. 그리고 모바일 GIS엔진의 구현은 C++표준 라이브러리와 추가 사용자 라이브러리를 이용하여 구현함으로써 이종 모바일 장치에 호환성 있는 모바일 GIS 엔진을 구현하였다. 또한 문자 인코딩 방식을 UTF-16으로 통일함으로써 문자처리에 대해서도 호환성을 유지할 수 있도록 구현하였다.

키워드

GIS 엔진, 모바일 장치, 모바일 GIS, 지리정보 시스템

1. 서 론

최근 스마트 폰이나 PDA와 같은 모바일 장치의 성능향상에 따라 이를 기반으로 하는 다양한 서비스 시스템이 연구·개발되어 지고 있다. 특히 GIS와 GPS를 이용한 차량용 네비게이션은 더 이상 연구수준이 아닌 상용 시스템으로 큰 성공을 이루었다.

최근에는 이러한 기술을 활용하여 사람이 도보 시 활용 가능한 네비게이션 시스템에 대한 연구가 진행되어지고 있다. 이는 기술적으로 큰 차이가 없어 보이지만, 정확도 측면과 시스템 의존성 측면에서 큰 차이를 보인다. 기존 차량용 네비게이션의 경우 차량별로 하드웨어를 구매하는 것이 당연시되어지고 있지만, 도보용의 경우 별도로 하드웨어를 구매하기에는 소비자 부담비용이 크다. 그러므로 기존에 보유하고 있는 모바일 장치를 활용한 방안이 필요하다. 따라서 본 연구에서는 시스템 의존성 측면에서의 문제점을 인식하고 이

를 해결할 수 있는 방법을 연구해 보았다.

일반적으로 모바일 장치는 장치마다 하드웨어 구성에 적지 않은 차이를 가진다. 이러한 차이에 따라 운영체제 역시 해당 하드웨어의 구성에 맞추어 재구성 한 뒤 포팅하게 된다. 응용 소프트웨어는 운영체제가 포팅 된 이후에 해당 모바일 장치에서 사용 가능한 라이브러리를 바탕으로 구현 되어진다. 따라서 응용 소프트웨어는 대부분 해당 모바일 장치에서만 구동되어지는 것이 일반적이다. 현재 구현되어져 있는 대부분의 모바일 장치 기반 GIS 서비스 시스템이 이러한 것이다.

본 연구에서는 이러한 문제의 해결을 위해 기존 시스템에서 모바일 장치에 독립적으로 구현 가능한 부분과 그렇지 않은 부분을 분리해 보았다. GIS 엔진의 경우 핵심 내용이 되는 데이터 로딩 및 관리 부분은 모바일 장치에 독립적일 수 있는 부분에 해당하고, 그래픽적인 부분은 모바일 장치에 독립적일 수 없다. 따라서 독립적일 수 있는 부분만을 객체(객체 라이브러리)로 설계하고, C++ 표준 라이브러리만을 이용하여 구현함으로써 다양한 모바일 장치에 호환성 있도록 하였다. 또한 GIS 엔진이 사용하는 문자 인코딩 방식을

***교신저자

UTF-16으로 통일함으로써 문자처리에 대해서도 호환성을 유지하면서도 다양한 언어를 지원할 수 있도록 구현하였다. 본 논문에서는 다양한 모바일 장치에서 간단한 GIS 뷰어를 구현함으로써 해당 GIS엔진의 호환성을 확인해 보았다.

II. 관련 연구

2.1 이종간 라이브러리의 차이점

C++ 언어는 표준 라이브러리가 존재하지만, 현재 각 벤더별로 제공하는 특정 컴파일 환경에서만 사용가능한 다양한 라이브러리를 제공하고 있다[1]. 그에 따라 특정 컴파일 환경에서의 라이브러리를 바탕으로 구현한 시스템은 다른 컴파일 환경의 모바일 장치에서는 호환되지 않는다. 또한 같은 컴파일 환경에서도 버전 업그레이드에 따라 기존에 사용하던 라이브러리가 없어지거나 새로이 추가됨에 따라 서로 간에 호환되지 않는 경우가 많다.

가장 대표적인 것이 문자를 저장하는 자료형이다. C언어에서는 1바이트의 char을 이용하였지만 이후 유니코드와 같이 멀티 바이트를 필요로 하는 문자코드가 정의되면서, 이를 위한 wchar_t 자료형이 추가되었다. 하지만 UTF-8, 16, 32와 같이 이를 지원하는 인코딩 형태가 다양하게 생겨나면서 각 컴파일 환경마다 서로 다른 형태를 사용하게 되었다. 현재 wchar_t 자료형에 대해 Windows CE는 2바이트로, iPhone에서 사용되는 MacOS 10에서는 4바이트로 정의하고 있다.

2.2 다양한 문자 인코딩

문자 인코딩의 방식은 대표적으로 아스키코드를 이용한 방식과 유니코드를 이용한 방식으로 나뉜다. 최근에는 다양한 언어를 지원하는 유니코드를 이용한 방식이 기본적인 방식으로 사용되어지고 있다. 하지만 동일한 유니코드를 인코딩하는 형태에도 UTF-8, 16, 32 등으로 다양하고 문자 인코딩 구조도 빅 엔디안, 리틀 엔디안으로 나뉜다[2].

기존 지리정보데이터의 경우 각 모바일 장치에서 사용되어지기 때문에 특정 형태 하나만 지정되어지면 그에 따라 시스템을 구현하기 때문에 문제가 되지 않았지만, 다양한 모바일 장치에서 사용하기 위해서는 이종의 모바일 장치에서 호환성 있게 사용할 수 있는 방식으로서의 통일이 필요하다. 또한 시스템 구현의 측면에서도 자원 제약이 심한 모바일 장치 환경에서 모든 경우에 대비해 비교하여 처리하는 모듈을 포함하는 것은 문제가 있다.

2.3 R-Tree 공간 인덱싱

R-Tree는 다차원의 공간 데이터를 인덱싱 하기 위한 자료구조이다[3,4,5]. 이 자료구조는 데이터를 최소 경계 사각형(MBR, Minimum Bounding Rectangle)들로 분할하여 저장한다. 각 MBR은 서로 겹칠 수도 있고 상위 레벨의 MBR은 하위 레벨의 MBR들을 포함하는 계층적인 트리 구조를 이룬다. R-Tree에서 자식 노드의 수는 미리 정해진 범위 내에서 변화한다. 이를 위해 각 노드는 생성 과정에서 재균형 과정을 반복하고, 완료된 트리는 균등한 크기의 자식 노드를 가지고 균등한 깊이로 구성되어지게 된다. 이를 통해 최종 R-Tree 생성에는 다소 긴 연산시간이 소요되지만 구성된 트리를 이용한 검색 시간은 감소시킬 수 있다.

III. 모바일 GIS 엔진 설계 및 구현

3.1 모바일 GIS 엔진 설계

모바일 GIS엔진은 전산화된 지리정보데이터를 파일이나 DB로부터 로딩하고, 로딩 된 데이터를 관리하는 기능을 한다. 또한 응용 시스템에서 데이터를 요청 시 필요한 데이터를 검색하여 반환하는 기능을 한다.

도형은 Point, MultiPoint, Polygon, Polyline 4개로 구성하였고, 해당 도형을 저장하고 관리하는 각 클래스를 설계하였다. 또한 유사한 데이터를 가지는 Polygon, Polyline은 ModelPolyShape 클래스로 동일 기능을 모으고, 데이터 로딩을 도와주는 PartIterator, PointIterator 인터페이스 클래스를 설계하였다. 이 클래스들은 GIS 엔진의 기본 자료형으로 사용한다. ShpModelLayer 클래스는 레이어를 관리하기 위한 클래스로 도형 타입과 파일이름을 가지고 데이터를 요청할 수 있는 인터페이스를 제공한다. 데이터 요청 인터페이스는 전체 데이터를 요청하는 것과, 전달된 MBR 영역 내부의 데이터만을 요청하는 두 가지 방식으로 설계하였다. 특정 MBR 영역에 대한 데이터

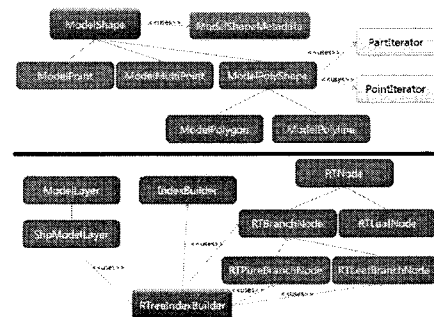


그림 1. GIS 엔진 클래스 설계도

요청은 RTreeIndexBuilder 클래스에서 미리 구성된 R-Tree index를 이용하여 실 도형 데이터를 로드하여 반환하도록 설계하였다.

그림 1은 본 연구에서 설계한 GIS 엔진의 클래스 설계도이다.

3.2 문자 인코딩 방식

문자 인코딩 방식은 UTF-16으로 통일하였다. 이는 Windows나 MAC 등의 컴파일 환경에서 최근 기본 문자 인코딩 방식으로 UTF-16을 사용하고 있기 때문이다. UTF-16은 유니코드를 문자코드 포인트로 사용하고 기본적으로 16비트로 하나의 문자를 표현하는 방식을 말한다. 본 연구에서 사용하는 모든 파일을 UTF-16으로 통일함으로써 서로 다른 컴파일 환경에서도 동일한 데이터로 인식할 수 있도록 하였다.

하지만 C++ 표준 라이브러리에서 멀티바이트 문자를 위한 자료형인 wchar_t에 대해 특정 벤드들이 먼저 라이브러리를 구축한 이후에 정의되어서 구체적으로 정의되지 않았다. 이에 따라 wchar_t 자료형에 대해 Windows에서는 2바이트, MAC에서는 4바이트로 인식하여 데이터 로드 과정에서 문제를 일으킨다. 본 연구에서는 이 문제를 해결하기 위해 GIS 엔진에 조건 컴파일을 사용하였다. MAC에서는 2바이트의 문자열 자료형으로 CFString을 제공한다. 따라서 Windows에서는 wchar_t와 wstring 자료형을 사용하고 MAC에서는 CFString을 사용하여 문자를 로드 할 수 있도록 하고, 이들 간의 선택은 조건 컴파일을 이용해 각 운영체제에 맞게 컴파일 하도록 하였다.

3.3 모바일 GIS 엔진 구현

본 연구에서 구현한 GIS 엔진은 기본 데이터 파일로 미국 ESRI사의 SHP 지리정보 데이터 파일을 기본으로 불필요한 데이터를 제거하고, 문자 인코딩 방법을 통일하여 새로운 형태의 파일 포맷을 정의하였다. 또한 데이터 저장 시 R-Tree 인덱싱을 통해 구성된 트리를 기본으로 하여 특정 크기의 상위 노드를 기준으로 정리하여 저장하였다. 이는 데이터 로드 시 특정 도형 하나씩을 로드하는 것이 아니라, 조건에 맞는 상위 노드가 검색될 경우 자식 노드 모두를 한꺼번에 로드하기 위함이다.

또한 한번 로드된 데이터는 캐시 메모리에 저장해 두었다가, 다음 검색 시 우선 검색하도록 하였다. 이는 지리정보 데이터 로드에서 완전히 다른 영역을 검색하는 경우 보다는 근처로 이동, 확대, 축소 작업이 많기 때문에 이러한 경우 다시 파일을 읽는 것 보다는 캐시 메모리에서 읽어 들이는 것이 효율적일 수 있기 때문이다.

그리고 모바일 GIS 엔진은 C++ 표준 라이브러리를 활용하고, 앞서 일부 부족한 부분을 보정하여 정적 라이브러리로 구현하였다. 이를 활용하

모바일 GIS 시스템을 개발할 때 일반 라이브러리와 같이 사용할 수 있고, 다양한 컴파일 환경에서 호환성 있게 구현하였다.

IV. 모바일 GIS 엔진을 이용한 뷰어 구현

4.1 GIS 뷰어 설계

GIS 뷰어는 본 연구에서 구현한 GIS 엔진의 활용 방안 제시 및 호환성 테스트를 위해 구현하였다. 기본적인 성능 테스트를 위한 것이기 때문에 GIS 뷰어의 다양한 기능 중에서도 가장 기본이 되는 확대, 축소, 전체보기, 이동 기능만을 구현하였다. 또한 여러 개의 레이어를 로드하기 위해 프로젝트 파일을 구성하였다. 프로젝트 파일은 txt파일로 레이어 이름, 파일경로, 도형타입, 출력 색상 및 형태(선 굵기)로 구성된다. 뷰어는 최초에 해당 프로젝트 파일을 로드하여 해당 레이어를 로드하고, 최대 MBR을 기준으로 레이어의 데이터를 로드해와 출력한다.

그림 2는 GIS 뷰어의 클래스 설계도이다. SC-LayerManager 클래스는 GIS엔진에서 로드한 레이어들의 화면 출력을 관리하고, 아래쪽에 있는 클래스들은 각 도형에 대한 실제 출력을 담당하는 클래스들이다.

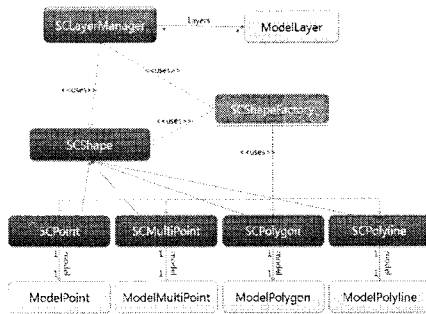


그림2. GIS 뷰어 클래스 설계도

4.2 GIS 뷰어 구현 결과

GIS 뷰어는 Window Mobile 버전과 MAC iPhone 버전으로 구현하였다. GIS 엔진을 추가 라이브러리로 포함하여 프로젝트를 생성하고, 앞서 설계에서 제안한 도형 출력 클래스와 레이어 관리 클래스를 구현하였다. 레이어 관리 클래스는 뷰어의 기본 기능인 확대, 축소, 이동 기능을 구현하여 정확한 GIS 데이터가 로드되어 출력되는지 확인하였다.

다음 그림3, 4는 각 운영체제 별 GIS 뷰어 실행 화면 이다.

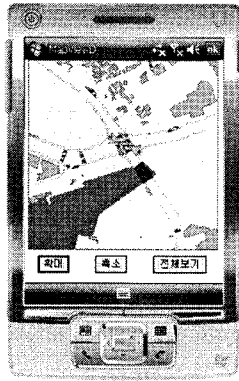


그림3. GIS View(Windows Mobile 6.0)



그림4. GIS View(MAC iPhone)

은 성능 최적화가 이루어지지 않았다. 이에 따라 특정 대용량의 데이터의 경우 데이터 처리에 다소 긴 시간이 소요되고 있다. 이를 최적화 하는 방법에 대한 연구도 병행하여 진행 할 예정이다.

참고문헌

- [1] 김승태, "표준 템플릿 라이브러리(STL, Standard Template Library)를 왜 사용하는가?", 2007, <http://webdizen.new21.net/blog/tag/Algorithm?TSESSION=13d103957b66f5776aca61293d00407e>
- [2] 한국어 위키백과, "문자 인코딩 [문자열 집합, 캐릭터 인코딩, 문자열 셋, Charset, 문자열 인코딩, 문자셋, 문자 집합, 인코딩], <http://enc.daum.net/dic100/contents.do?query1=10XXXXX536>
- [3] R-Tree 관련 도서자료(제목확인후 추가)
- [4] Antonin Guttman, "R-TREES A DYNAMIC INDEX STRUCTURE FOR SPATIAL SEARCHING"
- [5] D.M.Gavrila, "R-tree Index Optimization"

V. 결론

본 논문에서는 향후 모바일 장치 환경에서 지속적으로 개발되어질 GIS 서비스 시스템 개발을 위해 다양한 시스템에서 호환성 있게 작동할 수 있는 GIS 엔진을 설계·구현하였다. 본 논문에서 구현한 부분은 비록 GIS의 기본 기능만을 구현해 보았지만, 본 연구를 통해 기기 독립적인 모바일 GIS 엔진의 개발이 가능하다는 것을 확인할 수 있었다. 하지만 문자의 처리에 있어서는 아직 많은 부분에서 부족함을 확인할 수 있었다.

향후 연구에서는 기존 제작되어진 문자 자료형의 문제점을 재검토 하고, 이를 해결할 수 있는 방향을 모색해 볼 예정이다. 또한 현재 GIS 엔진