

# 모바일 벡터지도 서비스를 위한 클라이언트/서버 시스템의 설계 및 구현

김미란<sup>†</sup> · 최진오<sup>††</sup>

## 요약

휴대용 단말기를 이용한 모바일 벡터지도 서비스는 KVM(Kilobytes Virtual Machine)의 등장으로 인하여 PDA를 중심으로 보편화되고 있는 추세에 있다. 그러나, 휴대폰 단말기에서 벡터 지도서비스는 대역폭과 단말기의 자원 제약 등으로 인하여 구현이 쉽지 않은 문제점이 있다. 그러므로, 별도의 휴대폰 전용 지도 데이터베이스를 개발하여 서비스할 수 밖에 없다. 그러나, 간소화된 휴대폰 전용 지도 데이터베이스의 개발은 많은 비용을 초래할 뿐더러, 동일한 지역에 대한 다양한 사용자 뷰를 제공하기 어려운 문제점을 안고 있다. 따라서, 본 논문에서는 기존의 지도 데이터베이스를 그대로 이용하여 휴대폰에서의 모바일 벡터 지도서비스를 지원하는 JAVA 기반 서버/클라이언트 시스템을 새로 설계하고 구현하여 제시한다. 본 제안 시스템은 클라이언트 모듈의 기능을 최소화하고 서버에서 지도 간소화 작업을 수행하여, 기존의 지도 데이터베이스로부터 직접 모바일 벡터지도 서비스가 가능하게 하였다.

## Design and Implementation of Client/Server System for Mobile Vector Map Services

Mi-Ran Kim<sup>†</sup> · Jin-Oh Choi<sup>††</sup>

## ABSTRACT

The mobile vector map services for mobile devices, specially for PDA, are being popularized with the advent of KVM (Kilobytes Virtual Machine). However, the services for cellular phone have some problems for implementation, which are the restrictions of bandwidth, resource of the phone, etc. Thus, these restrictions force to develop new map database exclusively for cellular phone services. But, this approach needs high costs and could not support multi user view for a same area. Therefore, we suggest new client/server system for mobile vector map services, using existing general map database, and design and implement the system. The suggested system, which has minimized functions at client and processes map simplification works at server, makes it possible to serve mobile vector map without developing new map database only for cellular phone.

키워드 : 모바일 맵(Mobile Map), Generalization, Java, 벡터 지도(Vector Map), 휴대폰(Cellular Phone)

### 1. 서론

유선 인터넷 환경에서 제공되는 다양한 콘텐츠(contents)가 최근 무선 인터넷 환경으로 이동하고 있다. 그 중에서 전자지도 서비스에 대한 요구가 특히 증가하고 있다. 이동이 가능한 무선 단말기의 특성으로 사용자들의 지도 또는 약도 검색 요구가 많기 때문이다. 나아가서, 무선 전자지도 서비스는 특정 지역의 위치 검색 뿐만 아니라 GPS(Global Positioning System)와 연동한 개인 위치 추적, 물류시스템 관리, 관광지 안내 등 다양한 영역에 걸쳐 확산될 것으로 예

상되고 있다.

무선지도 서비스는 제공하는 데이터의 유형에 따라 래스터(raster) 방식과 벡터(vector) 방식의 두 종류로 나뉜다. 래스터 방식은 서버(server)에서 벡터 지도 데이터베이스로부터 이미지(image)를 동적으로 생성하거나 이미지 지도 데이터베이스로부터 질의 영역의 이미지를 그대로 클라이언트(client)로 전송하는 방법이다. 벡터 방식은 서버에서 벡터 데이터를 클라이언트로 그대로 전송하여, 클라이언트에서 이미지를 생성한다. 무선 지도 서비스에서 래스터 방식은 벡터 방식보다 상대적으로 서버가 전송해야 하는 데이터의 크기가 적으며 고정적이기 때문에 빠르고 일정한 대기 시간을 요구한다는 장점이 있다. 그러나, 지도의 출력 품질이 벡터 방식에 비해 떨어지고, 수신한 데이터가 이미

\* 본 연구는 정보통신부에서 지원하는 대학기초연구지원사업으로 수행되었음.  
† 준 회 원 : 부산외국어대학교 대학원 컴퓨터전자공학부  
†† 정 회 원 : 부산외국어대학교 컴퓨터전자공학부 교수  
논문접수 : 2002년 5월 21일, 심사완료 : 2002년 6월 20일

지이므로 벡터 방식처럼 다양한 용도로 이용할 수 없다는 단점을 가지고 있다. 이 논문은 휴대폰을 위한 벡터 방식의 무선 지도 서비스 기법을 논의 대상으로 한다.

무선 단말기는 크게 PDA(Personal Digital Assistants)와 휴대폰으로 나눌 수 있다. PDA는 휴대폰에 비하여 리소스의 용량이 크기 때문에 기존의 유선 인터넷을 통한 지도 서비스를 그대로 이용할 수 있다. 그러나, 휴대폰의 경우 출력 화면의 크기, 메모리 용량 등 리소스의 제약이 심하기 때문에 기존의 지도 서비스를 그대로 이용하기에는 많은 문제점이 있다.

그 동안 휴대폰 사업자와 일부 전문 업체들이 휴대폰에 전자지도도를 제공하는 제한적인 서비스를 시도하여왔다[3, 5, 16]. 그러나 여기에는 다음과 같은 문제점이 있다. 휴대폰을 위한 무선 환경의 지도 서비스를 위해 기존의 지도 데이터베이스를 직접 사용하기 어렵다. 휴대폰의 리소스 제약과 대역폭의 제한으로, 전송받은 데이터의 출력시 식별이 불가능하다. 또한, 한계를 넘어선 데이터 용량은 긴 응답시간을 필요로 할뿐만 아니라 클라이언트에서의 지도 출력 자체를 불가능하게 만든다. 이를 극복하기 위하여 휴대폰 전용 지도 데이터베이스를 별도로 구축하는 것은 추가적인 비용이 필요할 뿐만 아니라, 다양한 사용자의 뷰(view)를 지원하지 못하기 때문에 받아들일 수 없다. 즉, 데이터의 용량을 축소하여 구축한 데이터베이스는 특정 사용자의 특정 검색 요구에 응답하지 못할 수 있다는 것이다.

따라서, 기존의 일반적인 지도 데이터베이스로부터 직접 휴대폰을 위한 전자 지도 서비스가 가능하기 위해서는 첫째, 휴대폰의 작은 출력 화면에서 식별이 가능하도록 전송할 지도를 간략화시키는 동적인 기법이 필요하다. 둘째, 휴대폰의 리소스가 허용하는 용량 이내의 데이터만을 서버가 전송하는 동적인 기법이 필요하다. 셋째, 대역폭이 허용하는 용량 이내로 데이터를 축소시키는 기법이 필요하다.

본 논문은 기존의 지도 데이터베이스로부터 직접 모바일(mobile) 벡터 지도서비스가 가능한 시스템을 설계하고 이를 구현한 내용을 소개한다. 본 논문에서 설계하고 구현한 시스템은 Generalization 기법[10-13]을 이용하여 데이터를 간소화하고, 작은 출력 화면에서도 인식할 수 있는 지도를 생성해 내며, Filtering 기법을 이용하여 리소스와 대역폭의 허용 용량 이내로 데이터를 축소한다. 여기서 구현한 클라이언트/서버 시스템은, 서버 모듈은 JAVA 2 SE(Standard Edition)로 구현하였고 클라이언트 모듈은 JAVA 2 ME(Micro Edition)으로 구현하였다[1, 8]. 여기서, JAVA 기반으로 시스템을 설계하고 구현함으로써 얻을 수 있는 장점은, 첫째, 모바일 환경에서의 이식성 문제가 해결될 수 있다. 둘째, 기존의 WAP 기반 서비스에서 제한되었던 벡터지도 서비스가 가능하게 되었다[9, 14].

본 논문의 구성은 2장에서 관련 연구를, 3장에서 Generalization과 Filtering 개념을 설명한다. 시스템의 설계 내용

은 4장에서 소개하며, 구현 내용과 실험 결과는 5장에서 제시하고, 결론은 6장에서 맺는다.

## 2. 관련 연구

무선 지도 서비스의 구현은 WAP 기반과 JAVA 기반으로 나눌 수 있다. 최근에는 이식성, 동적 애플리케이션 다운로드의 장점, 벡터 지도의 구현으로 지도 품질 향상의 장점 등으로 인해 JAVA기반 서비스로의 이동이 가속화되고 있다[4, 5, 16].

본 논문은 JAVA 기반의 벡터지도도를 기존의 데이터베이스를 이용하여 휴대폰에 그대로 서비스하기 위하여 전통적인 Generalization 개념을 이용하였다. Generalization 과정은 일반적으로 Selection/Elimination, Simplification/Aggregation, Displacement/Symbolization 등의 세부 연산으로 구성된다[15, 10]. 지금까지 Generalization 연산들에 대한 분류와 정의는 개념적 수준에서 정립이 되어 있지만 세부 알고리즘은 상당한 노력과 전문적 판단을 필요로 하기 때문에 대부분의 관련 연구들은 Generalization의 여러 연산자 중에서 일부의 연산에 대한 세부 알고리즘을 제시하거나 규칙 베이스를 제시해 보이는 기초적 시도에 그치고 있다[11-13]. 본 논문에서는 휴대폰이라는 특수한 클라이언트 환경을 위해, 현재 인터넷을 통해 전자지도 서비스 중인 기존의 공간 데이터베이스([3])의 구조를 면밀히 파악하고 테스트하여, 위 Generalization 연산들의 알고리즘을 새로 개발하였다[6, 7].

무선 지도 서비스는 래스터와 벡터 지도 구분 없이 PDA를 통한 서비스는 일반적으로 확산되고 있는 추세이다[2, 3]. 특히 PDA를 이용하여 위치 추적을 통한 능동적 지도 서비스도 제공되고 있는 실정이다[3]. [2]의 연구에서는 무선 클라이언트인 PDA로 기존의 공간 데이터를 효율적으로 전송하기 위한 데이터 간소화 개념을 소개하였다. 그러나 이 연구는 PDA를 대상으로 하고 있으며, 제안한 방법이 단순한 Selection 연산에 그치고 있다.

휴대폰 단말기와 PDA 단말기와의 차이점은 출력 화면의 크기와 메모리 용량이다. 휴대폰의 경우 획기적인 데이터 간소화 작업 없이는 기존의 유선인터넷 환경에서 제공되는 양질의 공간 데이터베이스로부터 그대로 지도를 전송받아 출력할 수 없다. 본 논문은 이를 위하여 Generalization 연산 알고리즘을 휴대폰 환경에 적합하게 새로 개발하고 Filtering 처리 기법을 새로 고안하여, 큰 지도 품질의 손상 없이 제한된 리소스에서 지도 출력이 가능하도록 하였고 예측 가능한 응답시간을 보장하도록 하였다.

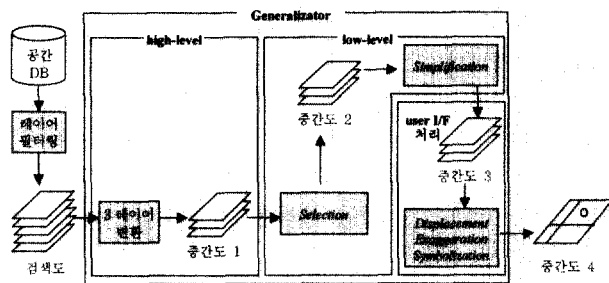
## 3. Generalization과 Filtering

지도의 정확성이 높아질수록 정보량이 많아지고 지도의 밀도가 높아진다. 이것은 휴대폰에서의 지도 인지도를 떨어

뜨리며, 과도한 데이터 크기로 인해 전송 지연이 발생하여 서비스가 불가능해진다. 따라서 정확성을 낮추고 허용되는 분량만큼의 정보만을 보낼 수 있는 기법이 필요하다. 또한, 무한정 데이터의 양을 줄일 경우 만족할 만한 정보를 얻지 못하게 된다. 따라서, 다음과 같은 두 가지 요소의 지원이 필요하다. 첫째, 원본 지도로부터 데이터 크기를 축소하되 필요한 정보가 누락되지 않도록 하는 데이터 축소 기준이 필요하다. 본 논문에서는 이 기준을 만족시키기 위해 Generalization 변환기법을 도입한다. 둘째, Generalization에 의해 데이터 크기가 축소된 지도라 할지라도 제한적인 대역폭의 임계치(threshold) 이상 또는 휴대폰의 허용 리소스 한계 이상의 데이터를 포함할 수 있다. 따라서 Generalization 결과가 허용치를 넘는 경우에는 다시 데이터의 크기를 강제로 줄여야 할 필요가 있다. 이 과정을 Filtering이라 부른다.

### 3.1 Generalization

(그림 1)은 본 논문에서 사용하는 Generalization 과정을 보인다. 이 과정은 다음과 같이 3 단계의 Generalization 과정을 통해 수행된다. 첫째, 우선 서비스용 검색도로부터 가상의 3 레이어 생성(중간도 1), 둘째 중간도 1로부터 각 가상 레이어(layer)별로 객체를 선택하거나 제거(중간도 2), 셋째, 복잡한 객체들을 단순화시키거나 복수개의 객체들을 통합하여 하나로 만들(중간도 3), 넷째, 지도의 인지도를 높이기 위해 데이터 크기의 증감 없이 객체를 변형한다(중간도 4). 첫 번째 과정을 본 논문에서는 high-level Generalization, 두 번째와 세 번째 과정을 low-level Generalization, 그리고 네 번째 과정을 사용자 인터페이스 처리 Generalization이라 부른다.



(그림 1) Generalization 과정

High-level Generalization 과정은 공간 데이터를 크게 3 레이어로 통합한다. 이것은 물리적 통합을 의미하는 것이 아니라 다수의 레이어를 가상의 3 레이어로 그룹화 시킨다는 것을 의미한다. 이 3 레이어는 사용자가 키워드로 검색하는 객체가 포함된 관심 레이어(interesting layer), 도로와 건물을 포함한 기본도 레이어(basic layer), 그리고, 나머지 모든 레이어는 마일스톤 레이어(milestone layer)로서 위치를 파악하는데 참고로만 이용한다. 이렇게 기존의 레이어들을 가상의 3 레이어로 매핑(mapping)하는 이유는 데이터에

대한 시맨틱(semantic)을 파악하여 Generalization에 활용하기 위함이다. 예를 들어 마일스톤 레이어에 대한 데이터 크기 축소는 심각한 정보 누락을 초래하지 않으므로 과감히 수행할 수 있으며, 관심 레이어에 대한 데이터 크기 축소는 검색 객체 하나만을 선택하면 된다. 즉, 가상 레이어별로 Generalization 기준을 달리 적용하여 효율을 높이기 위함이다.

Low-level Generalization 과정은 첫째, 전송하거나 출력하기에 너무 많은 객체를 포함한 경우 일부 객체만을 선택하거나, 또는 특정 레이어 전체를 생략하며(Selection), 둘째, 선택된 객체라 하더라도 지나치게 복잡할 경우 간단한 형태로 바꾸는 작업(Simplification)을 수행하는 것이다. 예를 들어, 가상 기본도 레이어에서 전송하고자 하는 영역 내에 건물 객체수가 너무 많은 경우, 일부만을 선택할 필요가 있다. 또한, 소축척 지도를 전송해야 할 경우, 도로 중앙선 레이어는 불필요한 데이터이므로 레이어 전체를 생략할 수 있다. 그리고, 도로를 표시하는 한 라인(line) 객체가 불필요하게 많은 포인트(point)로 구성된 경우(특히 소축척인 경우) 데이터 품질에 큰 차이 없이 많은 포인트를 생략할 수 있다. 또한, 복잡한 건물 모양을 그대로 전송할 필요 없이 규격화된 심벌만 전송하여 데이터의 전송량을 줄일 수 있다.

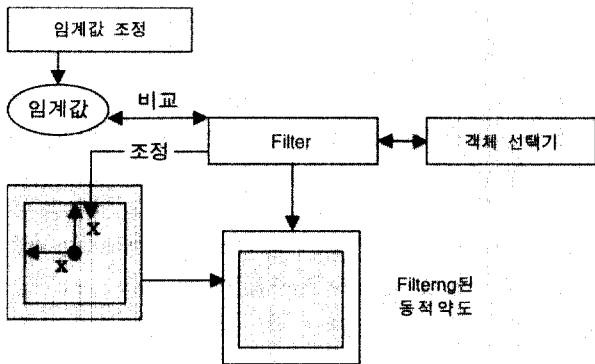
사용자 인터페이스 처리 Generalization 과정은 데이터 크기에 영향을 주지 않으며, 사용자의 지도 편독성을 높이기 위한 작업을 수행한다. Displacement 연산은 너무 인접하여 구분이 힘든 객체들을 분리하여 표기하는 작업이고, Exaggeration 연산은 중요 객체들을 크게 표현하거나 특정 색깔을 부여하여 눈에 띄게 만드는 것이며, Symbolization 연산은 특정 아이콘을 공간 객체 대신으로 대체하여 판별력을 높이는 작업이다.

여기서, 각 과정의 Generalization 연산들은 출력 축척에 따라 그 처리 메카니즘이 달라져야 한다. 예를 들어, Selection 연산은 대축척에서는 좁은 지역을 출력하기 때문에 많은 객체들이 선택될 수 있도록 하고, 소축척에서는 넓은 지역을 제한된 화면에 출력하기 때문에 상세 객체들은 선택하지 않도록 하여야 한다.

### 3.2 Filtering

Filtering은 Generalization을 거친 동적 약도가 클라이언트 휴대폰 단말기의 자원 한계내의 객체를 포함하고 있거나 통신 대역폭 이내의 객체를 포함하고 있을 경우에는 실시하지 않는다. 이 한계값을 임계값(threshold value)이라 부르며, 동적 약도가 이 임계값 이상의 객체를 포함할 경우에는 강제로 그 이하의 객체만을 포함하도록 객체를 삭제하는 것이 Filtering이다. Filtering을 통해 제거하는 객체는 키워드로 검색한 객체와의 거리 정보를 이용한다. (그림 2)는 Filtering 처리 과정을 보인다. 먼저, 객체 선택기는 사용자가 검색한 객체(출력 화면의 중앙에 위치)로부터 임의의 X값을 이용한 MBR(Minimum Bounding Rectangular)을 구

하고, 여기에 포함된 객체의 수를 구한다. 여기서, 객체 선택은 확률적으로 처리한다. 즉, MBR밖의 객체라도 낮은 확률로 선택될 수 있게 하여 지도의 품질을 높일 수 있다. 이렇게 선택된 객체의 수를 미리 설정된 임계값과 비교하여, 클 경우 X를 감소시키고, 작을 경우 증가시킨다. 이렇게 하여 임계값 이내의 최대 객체들을 포함하는 거리 X를 구한다. 즉, Filtering은 중심에서 멀어질수록 중요도가 떨어진다 고 판단한다.



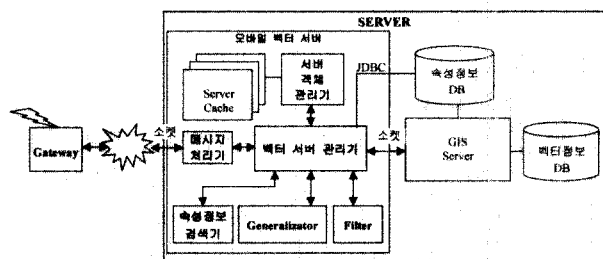
(그림 2) Filtering 과정

#### 4. 시스템 설계

##### 4.1 시스템의 구조

본 논문에서 설계하여 제시하는 모바일 벡터 전자지도 서비스를 위한 시스템 구조는 크게 두 모듈로 나뉜다. 첫째, 지도 데이터베이스로부터 벡터 데이터를 수집하여 클라이언트로 전송하는 모바일 벡터 서버 모듈, 둘째, 서버로부터 전송 받은 벡터 데이터를 클라이언트 단말기의 화면에 지도를 출력하는 모바일 벡터 클라이언트 모듈이다.

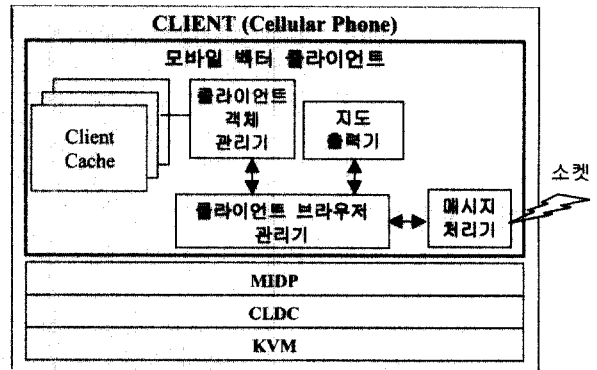
모바일 벡터 서버 모듈은, (그림 3)에서 보이듯이, 시설물 위치 데이터베이스(속성정보 DB)로부터 시설물을 검색하는 속성정보 검색기, 모바일 환경에 적합하도록 벡터 데이터의 양을 축소시키는 Generalizer, 차별적 응답 지연을 방지하



(그림 3) 모바일 벡터 서버 모듈

기 위한 Filter, 지도 데이터베이스로부터 수집한 벡터 데이터를 임시 저장하고 재사용하기 위한 서버 객체 관리자, 클라이언트와 통신을 처리하는 서버 메시지처리기, 그리고, 각 서브 모듈들을 통합 관리하는 벡터 서버 관리기로 나뉜다.

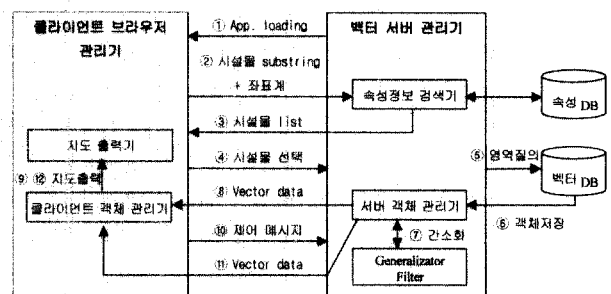
모바일 벡터 클라이언트 모듈은 서버로부터 전송받은 벡터 데이터를 임시 저장하고 재사용하기 위한 클라이언트 객체 관리자, 저장된 데이터를 휴대폰 출력 창에 지도로 그리는 지도 출력기, 서버와의 통신을 처리하는 클라이언트 메시지처리기, 그리고, 각 서브 모듈들의 통합 관리 기능을 하는 클라이언트 브라우저(browser) 관리기로 나뉜다. (그림 4)에서 이 모듈의 세부 구조를 보이고 있다.



(그림 4) 모바일 벡터 클라이언트 모듈

##### 4.2 시스템 처리 메커니즘

본 논문에서 설계한 모바일 벡터 서버 모듈과 클라이언트 모듈의 시스템 처리 메커니즘은 (그림 5)와 같다. 처음, 무선 단말기에 응용프로그램이 로딩(loading)이 되면(①), 클라이언트는 사용자가 검색하고자 하는 시설물 명과 단말기의 출력 화면 해상도를 서버로 전송한다(②). 서버는 속성정보 데이터베이스로부터 검색한 시설물 리스트를 클라이언트로 전송하여(③) 사용자가 정확히 검색하고자 하는 시설물 명을 넘겨 받는다(④). 서버는 지도 데이터베이스로부터 그 시설물이 포함된 영역의 벡터데이터를 질의하여(⑤) 객체로 저장한다(⑥). 그 다음, 지도 Generalization 작업과 Filtering 작업을 수행하여 벡터 데이터의 부륨을 축소시킨다(⑦). 이 결과 간소화된 벡터 데이터를 클라이언트로 전송한다(⑧). 이후 클라이언트는 사용자의 요구에 따라 제어 메시지를 서버로 전송하여 필요한 데이터를 추가로 전송 받는다(⑨⑩). 여기서 제어 메시지란 확대, 축소, 상하좌우 이동 등을 말한다.



(그림 5) 모바일 벡터 지도서비스를 위한 처리 메커니즘

### 4.3 메시지 구조

(그림 3)과 (그림 4)에서 보이는 서버와 클라이언트의 메시지 처리기는 처리 메커니즘에 따라 소켓으로 데이터를 주고 받기 위해 상호 메시지의 프로토콜을 정해 두어야 한다. (그림 6)은 (그림 5)의 처리 메커니즘 중에서 클라이언트가 서버에게 검색할 시설물명의 키워드를 전송하는 메시지 구조(그림 6) (가)와 서버가 벡터 데이터를 클라이언트로 전송할 때의 메시지 구조(그림 6) (나)를 예로서 보이고 있다. (그림 6) (나)는 포인트 데이터를 포함한 하나의 레이어(layer)만이 표현되어 있다.

Start	메시지 Type	검색어 bype	검색어	화면 width	화면 Height
5A16(1byte)	2 (1byte)	x (1byte)	(x 1byte)	w (1byte)	h (1byte)

(가) 검색할시설물명 전송(그림 5 ②)

Start	메시지 Type	검색어 bype	검색어	화면 width	화면 Height
5A16(1byte)	2 (1byte)	x (1byte)	(x 1byte)	w (1byte)	h (1byte)
시설물명 길이	시설물명	X 좌표	Y 좌표	Y-1개 순서쌍	X-1 순서쌍
y(1byte)	(y byte)	(1byte)	(1byte)	...	...

(나) 포인트 데이터 전송(그림 5 ⑧)

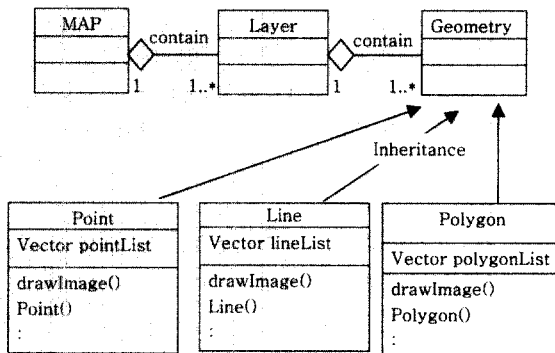
(그림 6) 메시지의 예

(그림 6) (가)에서 클라이언트가 서버로 휴대폰의 화면 해상도를 전송하는 이유는 좌표 변환을 서버에게 위임하기 위함이다. 지도 데이터베이스에서 사용하는 좌표는 보통 한 포인트 당 8바이트가 필요하다. 만약 클라이언트에서 좌표 변환을 수행한다면 한 포인트 당 8바이트를 전송받아 복잡한 실수 연산을 처리하여야 한다. 이것은 전송 비용이나 클라이언트의 계산 능력에 있어서 받아들이기 어렵다. 그래서, 서버에서 좌표변환을 수행하고 한 포인트 당 2바이트만 전송하는 기법을 채택하였다. 이렇게 함으로써 데이터를 압축하여 전송하는 효과를 얻을 수 있을 뿐만 아니라, 클라이언트에서 계산시간을 많이 소비하게 되는 압축 데이터의 해독 과정을 생략할 수도 있다.

### 4.4 저장 객체 모델

서버 객체 관리기와 질의한 벡터 데이터를 서버 캐쉬(cache)에, 클라이언트 객체 관리기는 서버로부터 전송 받은 벡터 데이터를 클라이언트 캐쉬에 각각 객체로 저장한다. 벡터 데이터의 임시 저장 객체 모델은 (그림 7)과 같다.

여기서, 객체들의 모음을 캐쉬라 부르는 이유는 이후 클라이언트의 제어 메시지에 따른 영역 질의를 벡터 서버로부터 재실행하지 않고 이미 검색되어 있는 객체는 재사용하는 정책을 이용하기 때문이다. 임시 저장된 객체의 재사용은 서버와 클라이언트에서 동시에 실시된다. 이에 대한 논의는 여기서 다루지 않는다.



(그림 7) 임시 저장 객체 모델의 UML

### 4.5 성능에 대한 논의

모바일 환경에서 벡터 전자지도 서비스 시스템을 설계하기 위해서 고려해야 할 사항은 다음과 같다. 첫째, 휴대폰과 같은 무선 단말기의 출력화면 크기제약으로 서버가 지도 데이터베이스에서 검색한 모든 데이터를 클라이언트로 전송할 수 없다. 무선 단말기에서 출력하더라도 해독이 불가능하게 복잡해지기 때문이다. 따라서 Generalization과 같은 데이터 간소화 작업을 서버에서 미리 수행할 필요가 있다.

둘째, 서버에서 무선 단말기로 전송하는 데이터 양을 일정하게 유지시킬 필요가 있다. 래스터 지도 서비스의 경우는 서버에서 클라이언트로 전송되는 데이터의 양이 일정하다. 그러나 벡터 데이터는 검색하는 지역에 따라 객체의 양이 상당히 다를 수 있다. 따라서 지나친 응답 지연시간의 차이를 막기 위해 비교적 일정한 양의 데이터로 벡터 데이터를 Filtering하여 클라이언트로 전송할 필요가 있다.

셋째, 무선 단말기의 리소스(resource)가 부족하기 때문에 지도를 출력하는 기능 이외의 대부분의 처리과정은 서버에서 수행되도록 하여야 한다. 예를 들어, 지도 데이터베이스에서 사용하는 좌표계는 휴대폰의 액정화면 좌표계로 변환되어야 하는데, 이 작업을 서버에서 처리한다면 효율이 향상될 수 있으며, 전송되는 데이터의 양도 줄일 수 있다. 이를 위해 (그림 5)의 ② 단계에서 무선단말기의 출력 해상도를 미리 서버가 전송받는 것이다.

넷째, 클라이언트에서 이전에 전송받았던 객체는 추가적인 무선 대역폭 사용을 줄이기 위해 재사용이 가능하여야 한다. 예를 들어 클라이언트에서 검색한 지도를 50% 스크롤(scroll)할 경우, 나머지 50%만 서버에 요청하여 응답 속도를 향상시킬 수 있다. 이를 위해서는 이미 전송 받은 객체들에 대한 캐싱(caching) 작업이 필요하다.

## 5. 구현 결과와 평가

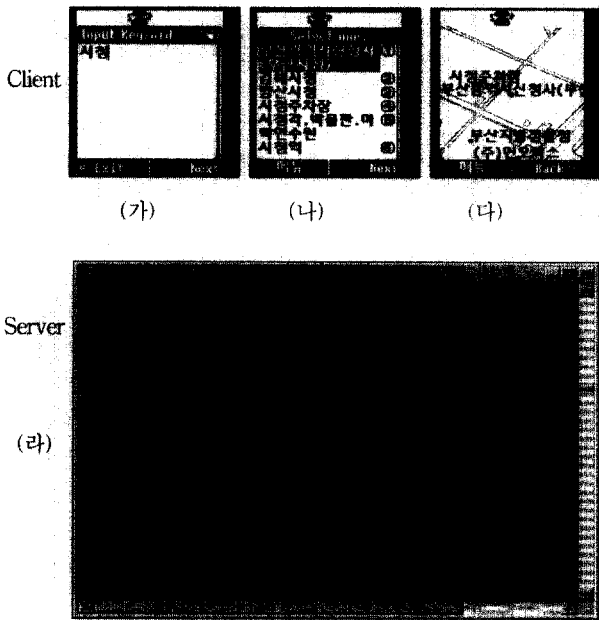
본 논문에서 구현을 위하여 사용한 환경은 다음과 같다.

- 서버 호스트 : Compaq Alphaserver DS10
- 지도 데이터베이스 : Cybermap Server Ver 2.0 ((주)사이버맵, 속성 데이터베이스 : Mysql DB)

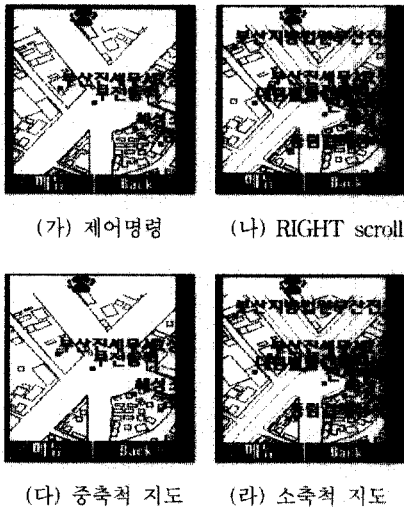
- 모바일 백터 서버 구현 : JAVA 2 SE, JDBC
- 모바일 백터 클라이언트 구현 : JAVA 2 ME
- 무선 단말기 : SK-VM Phone Emulator Ver 1.1  
(Samsung 전자, SCH-X230모델)

5.1 시설물 검색 과정

(그림 8)은 클라이언트에서 서버로 시설물 검색을 요청하고 결과 지도를 출력하는 과정을 보인다. (그림 8) (가)는 초기 모바일 백터 클라이언트 모듈이 클라이언트 단말기에 로딩된 후 사용자가 검색할 시설물명의 서브 스트링을 입력한 화면이다. (그림 8) (나)는 (그림 8) (가)에서 입력한 스트링이 서버로 전송이 되고 난 후 서버로부터 전송받은 시설물 리스트이다. (그림 8) (다)는 서버로부터 전송받은 백



(그림 8) 시설물 검색 과정



(가) 제어명령 (나) RIGHT scroll (다) 중축척 지도 (라) 소축척 지도

(그림 9) 제어 명령의 처리 화면

터 데이터를 클라이언트 모듈이 그린 결과이다. 그리고, (그림 9) (라)는 모바일 백터 서버 모듈의 실행 화면이다.

(그림 9) (가)는 확대, 축소, 또는 스크롤과 같은 제어 명령을 입력하는 클라이언트 화면이다. (나)는 (그림 8) (다)에서 오른쪽으로 스크롤하였을 때의 처리 화면을, (다)는 중축척 지도를, 그리고, (라)는 소축척 지도를 출력한 것이다. 대축척 지도는 (그림 8) (다)이며, 세로 240미터, 세로 140미터의 영역을 출력한다. 중축척과 소축척은 각각 대축척의 2배와 4배의 영역을 포함한다.

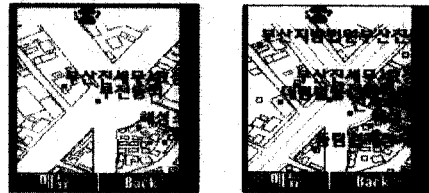
(그림 8)과 (그림 9)에서 구한 클라이언트의 지도 출력 화면은 출력이 가능하도록 Displacement 연산과 Filtering만을 수행한 결과로 얻은 것이다.

5.2 Generalization 처리 결과

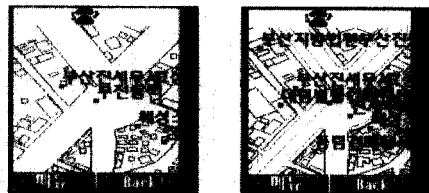
Generalization은 서버에서 처리하는 low-level Generalization과 클라이언트에서 처리하는 사용자 인터페이스 처리 Generalization으로 나뉜다. 서버에서는 Selection과 Simplification 연산을 처리하며, 클라이언트에서는 Displacement, Exaggeration, 그리고 Symbolization 연산을 처리한다.

(그림 10) (가)는 서버에서 Selection 연산을 거친 결과이고, (나)는 Simplification 연산을 거친 결과이며, (다)는 두 연산을 모두 거친 결과이다. (라)는 아무 연산도 거치지 않은 결과이다. (그림 10) (가)의 경우 불필요한 특정 레이어가 생략되었으며 가상 마일스톤 레이어도 5개로 제한되어 있음을 확인할 수 있다. 그리고, (나)의 경우, 복잡한 라인이나 폴리곤(polygon)들이 간략화되었다는 것을 확인할 수 있다. 여기서, 중축척과 소축척 결과를 출력하지 않은 이유는 (그림 10) (나)와 (다)의 경우는 경우에 따라 출력이 가능하지만 나머지 경우에는 클라이언트의 'Out of Memory'

출력레벨 : 대축척, 검색어 : 부전동역



(가) Selection 처리 (나) Simplification 처리



(다) 모두 처리 (라) Original Data

(그림 10) Low-level Generalization 처리

오류에 의해 비교 지도를 출력할 수 없기 때문에 생략한 것이다.

통계값을 구하기 위하여 본 논문에서 사용한 실험 방법은, 무작위로 추출한 지역에 대한 전송 포인트 개수를, 각 처리를 거친 경우와 거치지 않은 경우로 나누어 누적하였다. 그리고, 이를 여러 회 실시하여 반복 추출한 값과의 오차를 비교하였다. 이 오차는 반복 실험에서 1~2% 이내로 일정하였다. 그 이유는, 서버에서 전송하는 데이터가 Generalization 처리에 의해 객체가 일부 삭제되고 Filtering 처리에 의해 임계값 이내의 객체로 삭제된다. 여기서, 기존의 GIS 데이터베이스가 클라이언트에서 검색하고자 하는 영역 안에 포함하고 있는 공간 객체의 수는 대부분 무선 대역폭 이상의 객체이다. 따라서 Generalization과 Filtering을 적용할 경우 거의 일률적인 크기의 데이터가 클라이언트로 전송되며, 이 기법들을 적용하지 않을 경우에는 아예 휴대폰에서 지도 출력이 불가능하게 된다.

<표 1>은 서버가 클라이언트로 전송한 포인트 개수를 무작위 1000개 지역에 대하여 통계치로 구한 결과이다. <표 1>에서 Selection 연산에 의하여 평균 12.3%, Simplification 연산에 의하여 평균 46.3%, 모든 연산에 의하여 49.7% 데이터 전송량을 줄일 수 있음을 확인할 수 있다. 그리고, <표 1>에서 소축척에서 Generalization 효과가 중축척보다 더 크게 나타나는 이유는 소축척에서 Simplification 연산에 의해 수백개의 포인트로 이루어진 긴 라인이 수개에서 수십개로 축소될 가능성이 아주 높기 때문이다.

<표 1> Simplification 연산

출력 축척	Simplification 연산 처리(B)	Simplification 연산 처리(C)	모두 처리 (C)	모두 미처리 (D)	C/D
대	630	440	429	713	0.60
중	3372	2425	2378	3518	0.67
소	3425	1666	1439	4213	0.34

(무작위 1000개 지역에 대한 평균 전송 point 수)

(그림 10)의 지도 출력 결과를 보면 시설물의 텍스트 출력에 문제가 있음을 알 수 있다. (나)의 경우, 좁은 지역에 너무 많은 시설물이 선택되어 출력결과를 식별하기 어렵다. 따라서 Selection 연산에 의하여 우선순위(지명>관공서>거리이름>교통기관)에 의해 5개의 시설물만 선택한 것이 (그림 10) (다)의 결과이다. 그러나, 이것도 텍스트가 겹치거나 잘려서 식별이 어려운 문제를 여전히 안고 있다. (그림 11)은 클라이언트에서 처리하는 사용자 인터페이스 처리 Generalization을 거친 결과를 보인다. (그림 11) (가)에서 텍스트들은 서로 겹치거나 화면 경계 밖으로 잘리지 않고 적절한 위치에 출력되는 것을 확인할 수 있다. (그림 11) (나)는 시설물들의 레이어 종류에 따라 심볼이 추가된 결과이다. 그리고, (그림 11) (다)는 지도 위에 시설물 텍스트를 직접 출력하는 대신 번호 심볼로 출력하고 지도 아래에 별

도로 범례 처리한 결과를 보이고 있다.

출력조건 : 중축척, Generalization 수행



(가) Displacement (나) Exaggeration & (다) Symbolization Symbolization

(그림 11) 사용자 인터페이스 처리 Generalization

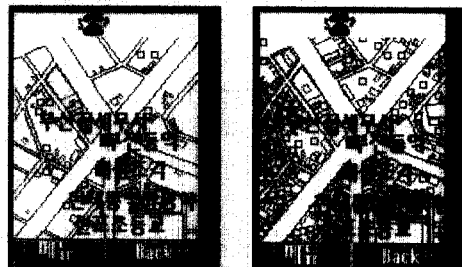
Generalization 처리 오버헤드는, 100개의 동일 지역에 대한 서버의 처리시간 A : (Generalization 처리에 걸리는 시간), 그리고, 처리시간 B : (Generalization을 수행하지 않을 때의 전체 처리시간)을 측정하여 비교한 결과(A/B), 서버의 전체 처리 시간을 약 20% 정도 증가시켰다. 이것은 Generalization 과정을 거치지 않을 경우 클라이언트에서 모든 데이터를 다 받기 위해 대기하는 시간과 비교하면 무시할 수 있다. 더구나, Generalization 과정을 거치지 않을 경우 많은 지역이 클라이언트에서 리소스 부족으로 출력되지 못한다.

### 5.3 Filtering 처리 결과

Generalization 과정을 거친 지도라도 경우에 따라 클라이언트에서 처리하지 못할 데이터 분량을 포함하고 있을 수 있다. 실험에 사용한 SCH-X230 단말기 모델은 약 2000개 포인트 이상의 데이터는 처리하지 못한다. 그리고, 약 1000개 포인트 이상의 데이터에 대해서는 급격한 처리 속도 감소를 보였다. 따라서, 데이터 임계치를 1000개 이내의 포인트로 제한하는 Filtering 처리를 통하여 서비스가 가능한 지도를 생성해내며, 클라이언트의 대기시간도 일정하게 유지시켜 주어야 한다.

(그림 12)는 Filtering 처리 결과를 보이고 있다. (그림 12) (가)는 Generalization 과정을 거친 지도가 1000 포인트 이상의 데이터를 포함하고 있어 Filtering 과정을 거쳐 1000개

출력조건 : 중축척, Generalization 수행



(가) point수 = 919 with Filtering (나) point수 = 2416 without Filtering

(그림 12) Filtering 처리 결과

포인트 이내의 지도로 변형한 결과이다. Filtering 처리는 화면 중심의 검색 시설물로부터 멀리 떨어진 객체일수록 높은 확률로 제거하며, threshold 값(실험에서는 1000개의 포인트) 이하의 객체만이 남을 때까지 중심쪽으로 접근하며 반복된다.

Filtering 처리 오버헤드는, 100개의 동일 지역에 대한 서버의 처리시간 A(Filtering 처리에 걸리는 시간), 그리고, 처리시간 B(Filtering을 수행하지 않을 때의 전체 처리시간)을 측정하여 비교한 결과(A/B), 서버의 전체 처리 시간을 약 30% 정도 증가시켰다.

### 6. 결론 및 향후 연구

본 논문은 휴대폰 등의 무선 단말기에 벡터 지도서비스를 위한 시스템을 설계하고 구현한 결과를 보였다. 이 시스템은 시설물과 벡터 데이터를 검색하고 가공 처리하여 클라이언트로 전송하는 서버 모듈과, 전송받은 벡터 데이터를 화면에 출력하는 클라이언트 모듈로 나뉜다.

본 논문에서 설계하고 구현하여 보인 제안 시스템의 특징은 다음과 같다. 첫째, 좌표변환, 데이터 압축 해독 등 클라이언트 모듈의 기능을 서버로 이동하고 최소화하여 제한된 리소스를 가진 휴대폰에서 벡터 데이터의 JAVA 브라우저를 구현할 수 있게 하였다. 둘째, 서버에서 지도 Generalization 작업을 수행함으로써, 새로운 휴대폰 전용 지도 데이터베이스의 개발 없이 모바일 벡터 전자지도 서비스를 가능하게 하였다. 셋째, 서버에서 지도 Filtering 작업을 수행함으로써 일정한 한계 이내의 클라이언트 대기시간을 보장하였다.

향후 연구되어야 할 내용은 래스터 지도서비스와의 성능 평가가 필요하다. 이것은 질적인 면과 성능 측면을 함께 비교 평가할 필요가 있다. 또한, 클라이언트 단말기의 리소스 제약을 극복하기 위해 클라이언트 모듈에서 객체 재사용을 위한 정책이 마련되어야 한다. 마지막으로, 기존의 벡터 지도 서비스 시스템과의 성능 비교 평가가 필요하다.

### 참 고 문 헌

[1] <http://java.sun.com/j2me/>.  
 [2] Y. S. Moon, K.-Y. K. Wong, "GSM Mobile Phone Based Communication of Multimedia Information : A Case Study," In Proc. 1st Int'l Conf. on MDA, pp.14-23, Hong Kong, China, December, 1999.  
 [3] <http://www.cybermap.co.kr/cm2000/newhome/index.html>.  
 [4] <http://developer.xce.co.kr/index.htm>.  
 [5] <http://www.forum.nokia.com/javaforum/>.  
 [6] J. Foley, A. Vandam, "Fundamentals of Interactive Computer Graphics," Addison Wesley press, 1984.

[7] S. Friedberg, A. Insel, L. Spence, "Linear Algebra," Prentice-Hall press, 1979.  
 [8] E. Giguere, "Java 2 Micro Edition," Wiley press, 2000.  
 [9] Wegdan Ahmad Elsay Fouda Abdelsalam, "Maintaining Quality of Service for Adaptive Mobile Map Clients," Master thesis of Mathematics in Computer Science in University of Waterloo, Canada, 2001.  
 [10] S. TAKINO, "GIS ON THEE FLY, To Realize Wireless Network by JAVA Mobile Phone," International Symposium on Asia GIS, 2001.  
 [11] J. C. Muller, J. P. Lagrange and R. Weibel, "GIS and Generalization : Methodology and Practice," Taylor & Francis, 1995.  
 [12] G. Robinson, F. Lee, "An automated generalization system for large scale topographic maps," Innovations in GIS1, pp. 53-63, 1994.  
 [13] G. Buddy, C. Jones, E. Furse, "A topological structure for the holistic generalization of large-scale cartographic data," Innovations in GIS2, pp.19-31, 1995.  
 [14] G. Dettori, E. Puppo, "Designing a Library to Support Model-Oriented Generalization," ACM, 1998.  
 [15] D. Horvat, D. Cvetkovic, V. Milutinovic, P. Kocovic, V. Kovacevic, "Mobile Agents and Java Mobile Agents Toolkits," International Conference on System Sciences, 2000.  
 [16] Weibel, R., "Summary report : Workshop on progress in automated map generalization," Technical Report ICA Working Group on Automated Map Generalization, Barcelona, September, 1995.  
 [17] <http://java.ez-i.co.kr/>.



### 김미란

e-mail : frankim@taejo.pufs.ac.kr

1996년 부산외국어대학교 컴퓨터공학과 졸업(학사)

1999년 신라대학교 교육대학원 컴퓨터 교육학과(교육학석사)

2000년~현재 부산외국어대학교 대학원 컴퓨터전자공학부 박사과정

관심분야 : 공학데이터베이스, 지리정보시스템, 무선 GIS 등



### 최진오

e-mail : jochoi@taejo.pufs.ac.kr

1991년 부산대학교 컴퓨터공학과 졸업(학사)

1995년 부산대학교 대학원 컴퓨터공학과(공학석사)

2000년 부산대학교 대학원 컴퓨터공학과(공학박사)

1998년~2000년 경동대학교 정보통신공학부 전임강사

2000년~현재 부산외국어대학교 컴퓨터전자공학부 조교수

관심분야 : 공학데이터베이스, 지리정보시스템, 모바일 GIS 등