

---

# 모바일 GIS를 위한 공간 데이터 간소화 기법에 대한 연구

최진오\*

A Study on Spatial Data Simplification Methods for Mobile GIS

Jin-Oh Choi\*

---

이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2002-003-D00278)

---

## 요 약

모바일(Mobile) GIS 서비스를 위해 전용 지도 데이터베이스를 별도로 구축하는 것은 막대한 중복 비용이 발생하므로 받아들이기 어렵다. 그러나 기존의 지도 데이터베이스를 그대로 모바일 GIS 서비스에 적용하는 것은 무선 대역폭, 단말기 자원(resources) 부족 등의 문제를 발생시킨다. 본 논문은 기존 지도 데이터베이스로부터 모바일 GIS 서비스가 가능하도록 하는 공간 데이터 간소화 기법을 제안한다. 제안하는 기법은 기존의 지도 일반화(generalization) 기법을 기반으로 모바일 환경에 맞도록 확장 구현된다. 본 논문에서 다루는 내용은, 지도 확대 축소에 따른 차별적 데이터 간소화 기법, 무선 단말기에 적합한 지도 출력 인터페이스 연구 등도 포함한다. 통신 지연, 처리 오버헤드 등 연구 결과의 평가가 실험을 통해 비교 분석된다.

## ABSTRACT

For the mobile GIS services, it is not acceptable to construct a new map database only for wireless service due to vast cost. But the existing map database cannot be used directly for the services due to narrow wireless bandwidth and shortage of mobile device resources. This thesis proposes spatial data simplification methods, thus, the existing map database enable the mobile GIS services. The proposing methods are based on the existing map generalization techniques. We extend it to mobile environments, and implement. This thesis also includes the issue of discriminative data simplification technique according to display level and map display interface suitable for mobile devices. Research results are estimated by experiments.

## 키워드

모바일 GIS 서비스, Generalization, 모바일 벡터 지도, 공간데이터 간소화

### 1. 서론

무선 통신 기술의 발전으로 유선 인터넷 환경에서 제공되던 다양한 콘텐츠(contents)가 무선 인터넷 환경으로 이동하고 있는 추세이다. 이 중 모바일 게임 등과 함께 각광받는 분야는 모바일

GIS 서비스이다. 이 분야는 단순한 전자 지도 서비스뿐만 아니라 지도를 기반으로 하는 GPS(Global Positioning System) 응용, LBS(Location Based Service) 응용 등 다양한 응용분야를 포함할 것으로 예상된다.

전자지도 서비스는 래스터(raster) 방식과 벡터

---

\* 부산외국어대학교 컴퓨터공학부 조교수

접수일자 : 2003. 12. 16

(vector) 방식으로 지원될 수 있으며, 모바일 환경에서도 이는 동일하다. 래스터 방식의 서비스는, 서버에서 검색한 전자지도를 이미지로 변환하고 WAP(Wireless Application Protocol)[1]을 통해 WML(Wireless Markup Language) 문서에 포함시켜 클라이언트로 전송하거나 J2ME(JAVA 2 Micro Edition)[2] 응용프로그램으로 전송하는 방식을 취한다. 벡터 방식의 서비스는, 서버에서 검색한 전자지도 데이터를 그대로 소켓을 통해 클라이언트로 전송하고, 클라이언트에서 수신한 공간 데이터를 J2ME 응용 프로그램으로 지도를 직접 그리는 방식을 취한다.

이러한 지도 기반 모바일 GIS 서비스의 지원에는 기존 유선 GIS 서비스에 사용되던 데이터베이스를 그대로 사용하기 어렵다는 문제점이 있다. 그 이유는 휴대폰으로 대표되는 모바일 클라이언트 장비는 출력화면, 자원, 대역폭 등의 제약으로 기존의 유선 서비스에서와 같은 품질의 지도를 출력할 수 없기 때문이다. 따라서 현재 대부분의 지도 콘텐츠 제공자들은 모바일 전용 GIS 데이터베이스를 새로 개발하거나 대축적으로 제한된 이미지 지도를 제공하고 있다.

첫째, 모바일 환경의 대역폭 제약으로 긴 지연 시간을 유발한다. 둘째, 휴대폰 단말기의 자원 제약으로 특정 크기 이상의 데이터를 수신할 수 없다. 셋째, 휴대폰 단말기의 작은 출력 화면으로 수신한 지도가 판독 불가능할 수 있다. 넷째, 휴대폰 단말기에서의 간소화 작업은 비효율적이다.

본 논문에서는 지도 일반화 기법[3]을 이용하여 서버에서 동적으로 지도 간소화를 수행하는 알고리즘을 모바일 환경에 맞도록 확장 개발하여 이러한 문제점을 해결하고자 한다. 또한, 출력 축척에 따른 알고리즘의 차별 적용으로 보다 세밀한 지도 출력 제어 기법과 모바일 단말기에 적합한 사용자 인터페이스를 제안한다. 제안 내용은 구현을 통하여 통신 지연 정도, 처리 오버헤드(overhead) 등을 실험 평가한다.

본 논문의 구성은 2장에서 관련 연구를 설명하며, 3장에서 본 논문의 제안 시스템 개요를 소개한다. 4장에서 제안하는 공간 데이터 간소화 기법을 설명하며 이의 구현 및 실험 평가 결과는 5장에서 소개한다. 결론은 6장에서 맺는다.

## II. 관련연구

지도 일반화 기법은 지도 제작 과정에서 사용자의 인지도를 높이기 위한 필수적인 요소이며, 전자 지도 서비스를 포함한 다양한 분야에서 자동 지도 생성을 위한 일반화 알고리즘들이 최근까지 연구되고 있다. 하지만, 일반화의 원칙들을 규정하는 명확한 법칙은 없다[4].

지도 일반화 기법에 대한 표준화 시도는, 1991년 ICA(International Cartographic Association) 산하의 지도 일반화 워킹 그룹이 구성되었고, 이후 지도 일반화 정형화 문제 등에 대한 연구 결과 [5][6]가 보고되었다. ESF(European Science Foundation)는 GISDATA라는 연구 프로그램에서 전자지도(digital map) 일반화의 현황을 연구하였다[3]. NCGIA(National Center for Geographic Information and Analysis)는 1993년 지도 일반화에 대한 전문가 모임(initiative 8 : Formalizing Cartographic Knowledge)을 구성하



그림 1. 판독 불가능한 지도

Fig. 1 Illegible Map

그림 1은 기존의 유선 GIS 데이터베이스로부터 검색한 지도를 휴대폰에 출력한 결과, 판독이 어려운 화면 예이다. 이렇게 기존의 유선 GIS 서버에서 검색된 모든 데이터를 휴대폰 단말기로 전송하는 것은 지도 판독 문제뿐만 아니라, 긴 지연 시간과 제한된 자원으로 인한 지도 출력 불가능 문제를 함께 발생시킨다.

따라서 서버에서 검색한 지도는 모바일 환경에 맞도록 간소화되어 클라이언트로 전송되어야 한다. 간소화가 반드시 필요한 이유는 다음과 같다.

고 연구 결과[7]를 발표하였다. 이 보고서에서 지도 일반화 및 전자지도 제작 논점들을 중점적으로 다루었다.

이러한 노력과 이후 관련 연구[8][9][10]들이 이어지고 있지만 일반화를 위한 많은 기본적인 문제점들이 아직 해결되지 못한 채 남아있다. 일반화 연산들에 대한 분류와 정의는 개념적 수준에서 정립이 되어 있지만 세부 알고리즘은 상당한 노력과 전문적 판단을 필요로 하기 때문이다.

무선 인터넷을 통한 전자지도 서비스 동향은 국내의 사이버맵[11], 포인트아이[12] 등의 상용 기술이 개발되어 있고, 국외의 경우, 일본의 NTT DOCOMO의 iMAP[13]과 네덜란드의 KPN 모바일의 지도 서비스[14]가 대표적이다. 그러나, 대부분 래스터 지도 서비스에 의존하고 있으며, 모바일 서비스를 위한 전용 지도 데이터베이스를 별도로 개발하여 사용하거나, 소축적으로의 지도 축소 기능을 지원하지 못하고 있다.

휴대폰에서 Java를 이용한 모바일 지도 서비스 구현에 대한 기초 기술 연구[15]와 PDA를 이용한 지도 간소화 시도[16]가 있었다. 그러나 본 논문에서 다루고자 하는, 기존 지도 데이터베이스로부터 지도 간소화 기법을 적용하여 휴대폰 모바일 GIS 서비스 지원 방법과는 구별된다.

### III. 제안 시스템 개요

모바일 GIS를 위한 공간 데이터 간소화 알고리즘을 개발하고 구현 실험하기 위해 본 논문에서 설계한 서버 시스템의 구조는 그림 2와 같다. 그림 2의 'Mobile Map Generator'는 GIS 서버로부터 검색한 지도를 'Generalizator'와 'Filter'를 통해 모바일 환경에 적합한 형태의 벡터 지도나 래스터 지도로 간소화하는 역할을 수행한다.

여기서 'Generalizator'는 일반화 연산들을 수행하여 지도를 간소화하며, 'Filter'는 대역폭 이상의 데이터를 포함하지 않도록 보장하는 역할을 담당한다.

모바일 클라이언트에서 래스터 지도를 출력하기 위해서는 서버에서 제공하는 지도 데이터의 형

식에 따라 내장된 WAP 브라우저(browser)를 이용하거나 J2ME 브라우저를 제작하여야 한다. 이 방식에서 지도 일반화는 데이터 크기를 줄인다고 보다는 지도 판독도를 높이는 역할을 한다.

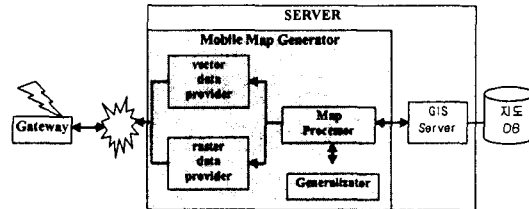


그림 2. 서버 시스템 구조  
Fig. 2 Server System Architecture

벡터 지도를 출력하기 위해서는 벡터 데이터 전송 프로토콜을 처리하고 수신한 데이터로 지도를 그리는 J2ME 브라우저를 반드시 개발하여야 한다. 본 논문에서 제안하는 지도 간소화 기법은 위 두 접근방법에 상관없이 적용될 수 있지만, 본 논문의 나머지 부분에서는 공간 데이터의 크기에 보다 민감한 벡터 방식 하나만을 대상으로 다루겠다.

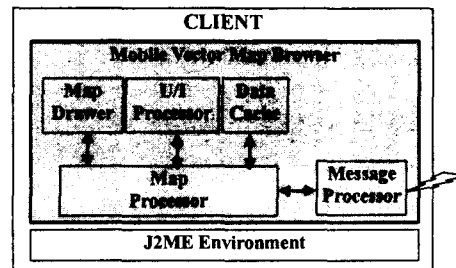


그림 3. 클라이언트 시스템 구조  
Fig. 3 Client System Architecture

그림 3은 벡터 방식을 지원하는 클라이언트의 브라우저 구조이다. 여기서 'U/I Processor'는 사용자가 지도를 쉽게 판독할 수 있도록 지도를 변형하기 위한 일반화 연산들을 수행한다. 그리고 'Data Cache'는 서버에서 전송받은 데이터를 필요에 따라 저장해 두었다가 추후 캐싱, 공간 분석 등에 활용할 수 있도록 지원한다. 'Map Drawer'는 전송받은 벡터 데이터를 그때 그때 캔버스에 그리거나, 'Data Cache'에 저장된 데이터를 이용

해 지도를 그리는 역할을 수행한다.

'Generalizator'의 상세 처리 과정은 다음과 같다. 먼저, 기존의 GIS 서버로부터 검색한 검색도를 지도 계층(layer)의 종류에 따라 가상의 3 계층 그룹(기본도 계층(Basic Layers), 검색 계층(Search Layers), 이정표 계층(Milestone Layers) 그룹)으로 나눈다. 그 이유는 이후의 일반화 연산에서 계층의 시맨틱(semantic)에 따라 간소화 기준을 달리 하기 위해서이다. 둘째, 저수준 일반화 과정에서 각 계층의 종류에 따라 selection, simplification 등의 일반화 연산들을 적용하여 공간 데이터를 간소화한다.

'Filter'의 상세 처리 과정은 다음과 같다. 먼저, 일반화 연산을 거친 지도의 크기를 측정하여 미리 설정해 놓은 임계값과 비교한다. 다음, 임계값을 초과할 경우, 미리 정해둔 우선순위에 의해 임의의 객체를 필요한 만큼 삭제한다.

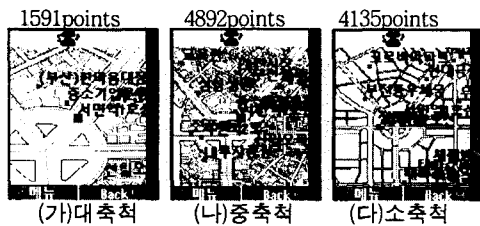


그림 4. 3단계 출력 축척  
Fig. 4 3 Level Display Scale

본 논문에서는 클라이언트의 지도 출력 축척에 따라 서버에서 일반화 연산을 차등 적용함을 실험하기 위하여 대축척(1:5000), 중축척(1:12000), 그리고 소축척(1:25000)의 3단계 출력 수준을 두었다. 그림 4는 각 축척으로 지도를 출력한 예이다 (이 출력 화면은 간소화작업을 거치지 않은 결과임).

#### IV. 공간 데이터 간소화 기법

공간 데이터의 간소화는 본 논문에서 제안하는 시스템의 'Generalizator', 'Filter', 그리고 'U/I Processor' 모듈들에 의해 수행된다. 그리고 각각의 세부 연산들은 출력 축척과 가상 계층 그룹에

따라 달리 수행된다. 예를 들어 selection 연산은 기본도 계층 그룹의 중축척 출력과, 이정표 계층 그룹의 대축척 출력을 위해 각각 다른 기준을 적용한다.

#### 1. selection 연산

selection 연산은 GIS 서버로부터 검색된 원본 공간 데이터에서, 출력되어야 할 객체들만 선택하고 나머지는 삭제(elimination)하는 일을 수행한다. 선택의 기준은 현재의 출력 축척과 가상 계층 그룹의 종류이다. 본 논문에서 제안하는 selection 알고리즘은 그림 5와 같다.

```

selection(map, scale, newMap) {
  for i<map.numOfLayers {
    if map.layer[i] is in BasicLayers[]
      if map.layer[i] is in preSet[sacle][B]
        addLayer(newMap, map.layer[i])
    else if map.layer[i] is SearchLayers[] {
      for j<map.layer[i].numOfObjects
        if map.layer[i].object[j] <> searchObject
          delObject(map.layer[i].j)
        addLayer(newMap, map.layer[i])
    }
    else if map.layer[i] is in MilestoneLayers[]
      if map.layer[i] is in preSet[scale][M] {
        leaveSomeObjects(map.layer[i],
numOfMSPerLayer[scale])
        addLayer(tempMap, map.layer[i])
      }
  }
  for k, total<numOfMSObjects[scale] {
    x=getPriority(tempMap, k)
    addLayer(newMap, tempMap.Layer[x])
    total=total+tempMap.Layer[x].numOfObjects
  }
}
    
```

그림 5. selection 알고리즘  
Fig. 5 selection Algorithm

이 알고리즘에서 'preSet[scale]'은 미리 결정한 특정 축척에서 출력할 계층들의 집합이다. 'searchObject'는 사용자가 검색하는 객체로서 누락되어서는 안되므로 반드시 포함시킨다. 'numOfMSPerLayer [scale]'은 현재 축척에서 각 이정표 계층에 포함시킬 객체의 최대 수를 지정한

것이다. 소축척일 경우 그 개수가 가장 많다. 'numOfMSObjects[scale]'은 축척에 따라 지도에 포함될 이정표 객체의 최대 수를 제한하기 위해 설정된 값이다. 역시 소축척일 경우 그 수가 가장 많다. 즉, 이정표 계층 그룹은 계층별로 특정 개수의 객체만 선택하고, 다시 전체 최대 허용 객체만큼만 우선순위 계층 순으로 채운다.

'getPriority()'는 이정표 계층들 중 주어진 우선순위의 계층 번호를 반환한다. 'delObject()'은 한 계층에서 주어진 한 객체를 삭제하는 함수이고, 'leaveSomeObjects()'는 한 계층에서 주어진 개수의 임의 객체만을 남기고 나머지는 삭제하는 함수이다.

## 2. simplification 연산

simplification 연산은 selection 연산에서 선택된 객체들의 구성 포인트 개수를 줄임으로써 공간 데이터 간소화를 수행한다. 그리고 simplification 연산은 각도 조사, 클리핑(clipping), 그리고 심블화(symbolization)의 3종류 세부 연산으로 구현된다.

```

simplification(map, scale) {
  for i<map.numOfLayers //degree check
    if map.layer[i].type=(LINE | POLYGON)
      for j<map.layer[i].numOfObjects
        for k<map.layer[i].object[j].numOfPoints
          if checkDegree(p[k],p[k+1],p[k+2],deg[scale])=DELETE
            delPoint(map.layer[i].object[j],k+1)
  for i<map.numOfLayers //clipping
    if map.layer[i].type=(LINE | POLYGON)
      for j<map.layer[i].numOfObjects
        for k<map.layer[i].object[j].numOfPoints
          if checkVisiable(p[k])=FALSE
            delPoint(map.layer[i].object[j],k)
  for i<map.numOfLayers //symbolization
    if map.layer[i] is in symTarget[]
      if map.layer[i].type=POLYGON
        for j<map.layer[i].numOfObjects {
          p=getCenterPoint(map.layer[i].object[j])
          delObject(map.layer[i],j)
          addPolyObject(map.layer[i],p.x,p.y)
        }
}
    
```

그림 6. simplification 알고리즘

Fig. 6 simplification Algorithm

각도 조사는 세 포인트를 조사하여 180도에 근접한 각도일 경우 중간 포인트를 삭제하는 것이다. 클리핑은 한 객체를 구성하는 포인트 중 화면의 뷰포인트(viewpoint)를 벗어나는 것들을 삭제하는 것이다. 끝으로 심블화는 복잡한 모양의 객체를 단순한 모양의 객체로 변형하는 것이다. 이 연산의 알고리즘은 그림 6과 같다.

이 알고리즘에서 'p[]'는 한 객체를 이루는 포인트이며, 'deg[scale]'은 미리 설정한 축척에 따른 임계 각도값이다. 이 값은 소축척일수록 커야 한다. 'symTarget[]'은 심블화할 대상 계층들을 미리 저장하고 있다. 그리고 'checkDegree()'는 주어진 3 포인트가 이루는 각도가 180도에서 임계값 내에 포함되는지 조사하며, 그럴 경우 'DELETE' 값을 반환한다. 'checkVisiable()'은 주어진 포인트가 뷰포인트 내에 포함되는 지 조사하며 그럴 경우 'TRUE'를 반환한다. 심블화 연산은 한 객체의 중심점을 구하고 삭제한 후 그 중심점을 가지는 간단한 폴리곤 객체를 추가하여 처리한다. 'addPolyObject()'가 이 역할을 수행한다.

## 3. filtering 연산

'Filter'에 의해 수행되는 filtering 연산은 객체를 선택한다는 점에서 selection 연산으로 볼 수 있지만, 저수준 일반화 과정이 모두 끝난 후 선택적으로 수행되기 때문에 별도의 일반화 연산으로 취급하기로 한다. 이 연산은 서버에서의 일반화 연산으로 간소화한 지도가 여전히 전송 대역폭 이상의 객체를 포함할 경우 강제로 특정 계층의 특정 객체들을 삭제하는 것이다. 삭제 대상 계층은 지도 관독에 큰 지장을 주지 않으면서 객체 수가 많은 것을 택하며(예:가옥 계층), 삭제 대상 객체는 검색 객체에서 먼 거리에 있는 것을 높은 확률로 택한다. 이 알고리즘은 그림 7과 같다.

여기서, 'filterThreshold'는 허용 임계 포인트 개수이며, 전송 지연시간과 휴대폰 단말기의 최소 메모리 용량을 고려하여 결정한다. 'filterTarget[]'은 필터링할 대상 계층들을 미리 저장하고 있다. 그리고 'getDistance()'는 주어진 객체와 검색 객

체와의 거리를 반환하며, 'getDelProb()'는 주어진 거리값이 클수록 높은 확률로 'TRUE' 값을 반환한다.

```
filtering(map) {
    for countPoints(map) > filterThreshold
    for i<map.numOfLayers
        if map.layer[i] is in filterTarget[]
            if map.layer[i].type=POLYGON
                for j<map.layer[i].numOfObjects {
                    dist=getDistance(map.layer[i].object[j])
                    if getDelProb(dist)=TRUE
                        delObject(map.layer[i].j)
                }
    }
```

그림 7. filtering 알고리즘  
Fig. 7 filtering Algorithm

4. displacement 연산

'displacement'는 지도 간소화 작업보다는 판독도를 높이기 위해 객체들의 위치를 임의로 이동시키는 연산을 말한다. 그러나 본 연구에서는 이정표 계층 객체들의 속성정보 출력에 이 연산을 적용하는 것이 오히려 휴대장치에서 지도 판독도 향상에 큰 도움이 된다는 것을 실험에서 고찰할 수 있었다. 따라서 이 연산은 객체의 좌표 이동보다는 속성정보 출력 위치 이동을 대상으로 한다. 이 연산의 알고리즘은 그림 8과 같다.

```
displacement(map) {
    for i<map.numOfLayers
        if map.layer[i] is MilestoneLayers[]
            for j<map.layer[i].numOfObjects
                addObject(msLayer, map.layer[i].object[j])
    sortByYCORD(msLayer)
    for 0<=i<msLayer.numOfObjects/2
        p=findUpBestPoint(i,msLayer)
    for msLayer.numOfObjects>i>=msLayer.numOf-
    Objects/2
        p=findDownBestPoint(i,msLayer)
    }
```

그림 8. displacement 알고리즘  
Fig. 8 displacement 알고리즘

이 알고리즘은 먼저 이정표 객체들만 'msLayer'

로 모아서 y좌표 기준으로 정렬한다. 그리고 상위 절반은 'findUpBestPoint()'로 위에서부터 최적의 속성정보 출력 위치를 찾는다. 이것은 출력될 속성정보가 화면 바깥으로 잘려나가지도 않고, 이미 결정된 영역과 충돌도 없는 시작점과 문자가 차지할 영역을 찾는 것이다. 'findDownBestPoint()'는 충돌 확률을 줄이기 위해 가장 아래에서부터 하위 영역을 우선 탐색한다.

5. symbolization 연산

이 연산은 이정표 객체의 출력을 심볼화하는 것을 말한다. 심볼화는 공간 데이터 대신 해당 계층의 범례를 출력하는 방법과 번호를 매기고 속성정보를 범례 처리하는 방법으로 나눌 수 있다. 이를 구현한 예는 그림 9와 같다.

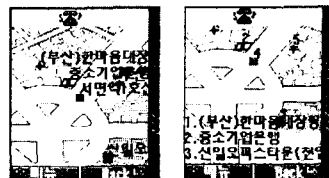


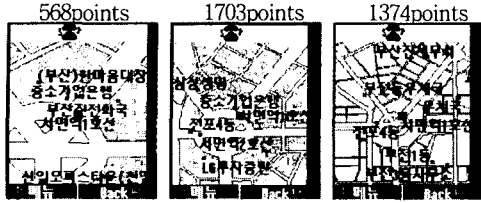
그림 9. symbolization 구현 예  
Fig. 9 symbolization Implementation Example

V. 구현 및 실험 평가

이 장에서는 제안한 공간 데이터 간소화 기법을 구현하여 실험 평가한 결과를 보인다. 구현 환경은 다음과 같다.

- 서버 호스트 : Compaq Alphaserver DS10
  - 지도 DB : Cybermap Server Ver 2.0
  - 속성 DB : Mysql
  - 모바일 백터 서버 구현 : JAVA 2 SE, JDBC
  - 모바일 백터 클라이언트 구현 : JAVA 2 ME
  - 무선 단말기 : SK-VM Phone Emulator Ver 1.1 (SCH-X230)
  - 실제 휴대폰 실험 : 삼성전자 SCH-X570
- 구현 결과 지도는 그림 10과 같다. 그림 4의 간소화 작업을 거치지 않은 지도와 비교하면 서버에서 클라이언트로 전송되는 데이터 크기가 대폭 줄

고 지도 판독도도 향상되었다는 것을 알 수 있다.



(가)대축척 (나)중축척 (다)소축척

그림 10. 간소화 결과 화면  
Fig. 10 Simplification Result Maps

그림 11은 일반화 연산의 데이터 간소화 효과를 보인다. 이 실험 결과는 서버가 클라이언트로 전송한 포인트 개수를 무작위 1000개 지역에 대하여 평균값으로 구한 것이다. 이 결과에서 간소화 작업에 의해 대축척은 평균 49%, 중축척은 69%, 그리고 소축척은 85% 데이터 전송량을 줄일 수 있음을 확인하였다.

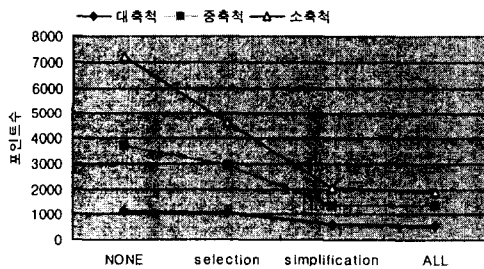


그림 11. 데이터 간소화 효과  
Fig. 11 Data Simplification Effect

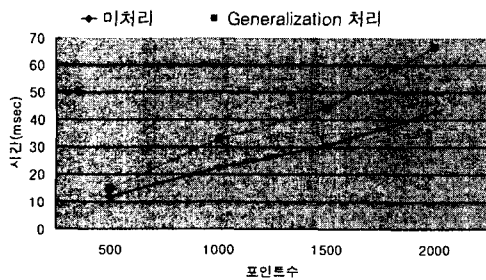


그림 12. 데이터 간소화 처리 오버헤드  
Fig. 12 Data Simplification Processing Overhead

그림 12는 일반화 연산 및 필터링 처리의 오버

헤드를 무작위 1000개 지역에 대한 지도의 전송 포인트 수와 서버에서의 처리 시간을 평균하여 비교한 것이다. 간소화 처리 오버헤드는 포인트가 증가함에 따라 늘어나지만 그 차이는 간소화 효과와 비교한다면 무시할 수 있는 정도이다.

그림 13은 서버에서 전송하는 포인트 수에 따른 클라이언트에서의 지연 시간을 비교한 실험 결과이다. 전송 지연 시간은 1000 포인트를 기준으로 급격히 증가한다. 이는 휴대폰에서의 메모리 부족으로 인한 처리 지연으로 판단된다. 이 통신 지연시간은 그때 그때의 통신망 상태에 따라 가변적일 수 있지만, 근본적으로는 전송되는 포인트 수에만 관련이 있다. 이 실험을 통하여 모바일 환경에서 공간 데이터 간소화의 필요성을 확인할 수 있다.

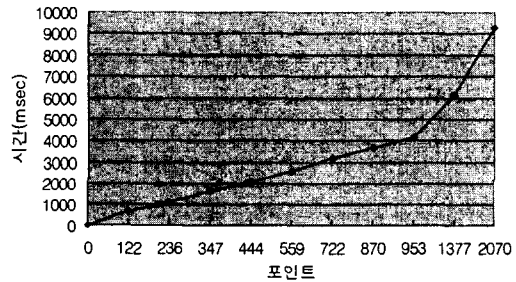


그림 13. 전송 지연 시간  
Fig. 13 Transmission Delay Time

## VI. 결론

본 논문은 모바일 환경에서 전자 지도 서비스를 하는데 있어서 발생하는 문제점을 지적하였고, 이를 극복하기 위한 공간 데이터 간소화 알고리즘을 소개하였다. 또한, 이 알고리즘을 구현 실험하기 위한 시스템 구조를 제안하였다. 제안 시스템은 구현하여 그 성능을 실험 평가해 보았다.

본 논문에서 제안하는 공간 데이터 간소화 알고리즘은 선행 연구들에서 제안되었던 기법들에 기초하여 모바일 환경에 적합하도록 실험을 반복하여 확장 개발한 것이다. 제안한 알고리즘은 지도의 계층 종류별로 3개의 가상 계층 그룹으로 나누고, 이 그룹과 출력 축척에 따라 저수준 일반화

연산 기준을 달리 적용하는 기법을 사용하였다. 그 결과, 데이터 간소화에 수반되는 정보 손실을 최소화할 수 있었으며, 휴대폰에서의 소축적 지도 출력도 가능하게 하였다.

데이터 간소화 알고리즘의 구현 결과는 벡터 지도 서비스의 휴대폰 출력 화면으로 보였고, 데이터 간소화 효과, 처리 오버헤드, 그리고 통신 지연시간을 실험을 통하여 수치화하고 분석하였다. 결론적으로, 데이터 간소화 효과는 지도의 중요 정보 손실 없이 상당히 크고, 처리 오버헤드는 그 효과에 비하여 크지 않은 것을 알 수 있었다.

### 참고 문헌

[1] "WAP Forum Specifications", WAP June 2000 Conformance Release, 2000  
 [2] <http://java.sun.com/j2me/>  
 [3] Muller, J.C., Lagrange, J. P. and Weibel, "GIS and Generalization : Methodology and Practice", Taylor & Francis, 1995  
 [4] ESRI, "Automation of Map Generalization : The cutting-Edge Technology", An ESRI White Paper, Environmental Systems Research Institute, Inc., 1996  
 [5] Weibel, R., "Summary report : Workshop on progress in automated map generalization", Technical Report ICA Working Group on Automated Map Generalization, Barcelona, September, 1995  
 [6] Mackaness, W. A. and Weibel, "Report of the ICA Workshop on Map Generalization", Technical Report ICA Working Group on Automated Map Generalization, Sweden, June, 1997  
 [7] Barbara P. Buttenfield, Catherine Dibble, "Research Initiative 8 Formalizing Cartographic Knowledge: Scientific Report for the Specialist Meeting", NCGIA, 1995  
 [8] Ruas, Anne, "A method for building displacement in automated map generalization", International Journal of Geographical Information Science, Vol. 12, No. 8, pp. 789-803, 1998  
 [9] Dettori, G. and Puppo, E., "Designing a Library to Support Model-Oriented Gen-

eralization", ACM Int. Symp. on Advances in GIS, United States, 1998

[10] ESRI, "Map Generalization in GIS : Practical Solutions with Workstation ArcInfo Software", An ESRI White Paper, Environmental Systems Research Institute, Inc., 2000  
 [11] <http://www.cybermap.co.kr/cm2000/newhome/products/phone.html>  
 [12] [http://www.pointi.co.kr/service/sv03\\_01.html](http://www.pointi.co.kr/service/sv03_01.html)  
 [13] [http://www.nttdocomo.co.jp/english/p\\_s/i/iaera/main.html](http://www.nttdocomo.co.jp/english/p_s/i/iaera/main.html)  
 [14] <http://www.viamichelin.com/viamichelin/gbr/jsp/num/gen/html/MaProducts>  
 [15] Takino, S., "GIS On The Fly, To Realize Wireless Network by JAVA Mobile Phone", Int. Symp. on Asia GIS, Japan, 2001  
 [16] Jane Liu, "Mobile Map: A Case Study in the Design and Implementation of a Mobile Application", Master Thesis of Carleton University, Canada, 2002

### 저자 소개



#### 최진오(Jin-Oh Choi)

1991년 부산대학교 컴퓨터공학과 공학사  
 1995년 부산대학교 대학원 컴퓨터공학과 공학석사

2000년 부산대학교 대학원 컴퓨터공학과 공학박사  
 1998. 3 ~ 2000. 2 경동대학교 정보통신공학부 전임 강사

2000. 3 ~ 현재 부산외국어대학교 컴퓨터공학부 조교수

※ 관심분야 : 공학데이터베이스, 지리정보시스템, 모바일 GIS