

모바일 벡터 지도 서비스를 위한 객체 재사용 기법의 설계 및 구현

김진덕[†] · 최진오^{††}

요 약

클라이언트에서 검색한 지도를 스크롤할 경우 이미 전송받은 객체를 캐싱하여 무선 데이터 전송량을 줄일 수 있지만, 클라이언트 내에서 데이터 좌표 변환, 선택적인 객체 삭제, 압축 단계, 구조화 단계가 부가적으로 요구된다. 특히 좌표 변환 연산은 모바일 휴대폰의 낮은 계산 능력, 적은 메모리 용량 등과 같은 제한적 자원으로 많은 시간이 요구된다. 따라서 휴대폰 기반 모바일 지도 서비스에서 효과적인 지도 제어 연산을 수행하기 위해서는 휴대폰의 제한적인 자원을 극복함과 동시에 무선 네트워크 대역폭을 줄이는 방안으로서 객체 재사용 기법이 연구되어야 한다. 이 논문에서는 클라이언트-서버 기반 모바일 벡터 지도 서비스 시스템에서 지도 제어를 위한 객체의 효율적인 객체 재사용 기법을 제안한다. 또한 실제 데이터를 제공하는 Web GIS 시스템을 이용한 실험 평가는 제안한 기법이 모바일 폰을 위한 지도 서비스에 적합하다는 것을 보여준다. 그리고 실험 결과를 토대로 객체 재사용 방법과 재전송 방법의 장단점을 분석한다.

Design and Implementation of Object Reusing Methods for Mobile Vector Map Services

Jin Deog Kim[†] · Jin Oh Choi^{††}

ABSTRACT

Although the reuse of the cached data for scrolling the map reduces the amount of passed data between client and server, it needs the conversions of data coordinates, selective deletion of objects, cache compaction and object structuring step in the clients. The conversion is a time-intensive operation due to limited resources of mobile phones such as low computing power, small memory. Therefore, in order to control the map efficiently in the vector map service based mobile phones, it is necessary to study the methods which reuse cached objects for reducing wireless network bandwidth and overwhelming the limited resources of mobile phones as well. This paper proposes the methods of reusing pre-received spatial objects for map control in the mobile vector map service system based on client-server architecture. The experiments conducted on the Web GIS systems with real data show that the proposed method is appropriate to map services for mobile phone. We also analyze the advantages and drawbacks between the reuse of cached data and transmission of raw data respectively.

키워드 : GIS, 모바일 벡터지도 서비스(Mobile Vector Map Service), 캐쉬(Cache), J2ME

1. 서 론

최근 무선 인터넷의 급속한 확산과 휴대장치 기술의 발전으로 모바일(Mobile) 지도 서비스가 보편화되고 있다. 이동이 자유로운 휴대 장치는 향후 정보 소비의 주체가 된다는 점을 감안할 때 모바일 지도 서비스는 GPS(Global Positioning System)와 연동한 위치 기반 시스템(LBS : Location Based System)[5], 물류 관리 시스템, 관광지 안내 등 많은 응용 분야를 창출할 것이다[2]. 특히 보급률이 매

우 높은 휴대폰에서의 지도 서비스는 대표적인 콘텐츠가 될 것이다.

무선 인터넷 응용의 구현은 크게 WAP 기반과 JAVA 기반으로 나누어진다. 최근에는 이식성과 동적 애플리케이션 다운로드의 장점으로 인해 JAVA의 J2ME가 많이 사용되고 있다[11]. 그러나 지금까지 휴대폰 기반 모바일 지도 서비스는 WAP 기반 시스템으로서 비교적 간단한 구조인 래스터(Raster) 방식의 공간 데이터 구조를 이용한다. 그렇지만 래스터 데이터는 일반적으로 벡터(Vector) 데이터에 비해 용량이 클 뿐만 아니라 축소 확대 및 질적 저하가 발생한다[4].

또한 모바일 지도 서비스는 위치 기반 시스템(Location Based System)[12]이 다양하게 개발되었으며, 최근에는 GPS

※ 본 연구는 정보통신부에서 지원하는 대학기초연구지원사업으로 수행되었음.
[†] 정 회원 : 동의대학교 컴퓨터공학과 교수
^{††} 정 회원 : 부산외국어대학교 컴퓨터공학과 교수
 논문접수 : 2002년 7월 4일, 심사완료 : 2001년 1월 27일

와 연동하여 위치 추적이 가능한 제품 또한 개발되었지만 주로 PDA 응용 프로그램에서 서비스되고 있다. 본 논문에서 다루는 시스템 구조는 휴대폰과 같이 제한된 저장 용량, 낮은 계산 능력, JAVA 기반이라는 면에서 PDA와는 하드웨어와 소프트웨어적인 측면에서 많은 차이가 있다.

이 논문에서 구현한 시스템은 서버와 클라이언트 모듈로 나뉜다. 서버 모듈은 JAVA 2 SE(Standard Edition)로 구현하였고 클라이언트 모듈은 JAVA 2 ME(Micro Edition)으로 구현하였다[11]. 여기서, JAVA 기반으로 시스템을 설계하고 구현함으로써 얻을 수 있는 장점은, 첫째, 모바일 환경에서의 이식성 문제가 해결될 수 있다. 둘째, 기존의 WAP 기반 서비스에서 제약되었던 벡터지도 서비스가 가능하게 되었다.

한편 GIS 응용 프로그램에서는 스크롤, 축소, 확대와 같은 지도 제어(Map Control) 연산이 빈번하게 발생한다. 래스터 데이터는 데이터 구조적인 측면에서 각 객체들을 독자적으로 관리하거나 처리할 수 없지만, 벡터 데이터 구조는 각 객체별로 처리가 가능하다. 그러므로 지도 제어 시 벡터 데이터의 객체 처리방법은 전체 객체를 전송 받는 방법과 이미 전송 받은 객체를 캐싱하여 재사용하는 방법을 고려해 볼 수 있다.

물론 고성능 PC를 클라이언트로 활용하는 시스템 구조에서는 객체를 캐싱하여 재사용하는 것이 당연히 좋은 성능을 보인다. 그러나 이 논문의 환경인 휴대폰을 클라이언트로 이용하는 시스템 구조에서는 휴대폰의 낮은 계산 능력과 무선 데이터 송수신이라는 점 때문에 객체 재사용 방법이 항상 좋은 성능을 보인다는 보장이 없다. 실제로 캐싱된 데이터의 재사용을 위해서는 클라이언트에서 기존 데이터의 선택적인 삭제, 좌표변환, 메모리 압축, 새로운 데이터의 수신 및 객체 구조화 과정이 요구된다[6, 7]. 따라서 휴대폰을 이용한 지도 제어시 단순 객체 전송 방법과 캐싱된 객체의 재사용 방법의 성능은 다양한 실험을 통해 수행 시간을 기준으로 평가할 수 밖에 없다.

따라서 이 논문에서는 모바일 벡터 지도 서비스 시스템을 위한 효율적인 객체 재사용 기법을 제안하고자 한다. 제안하는 기법은 불특정 다수가 동시다발적으로 접근하는 서버의 부하를 줄이고, 무선 데이터 전송량을 줄이며[8, 9], 휴대폰의 낮은 계산 능력을 감안하여 효과적으로 지도 제어를 하고자 함에 목적이 있다. 이를 위해 실제 데이터를 제공하는 Web GIS 엔진을 서버로 이용하며, 클라이언트로 휴대폰을 이용하는 시스템을 구축하고, 실험을 수행하여 캐싱된 데이터의 재사용과 객체 재전송 방법의 선택 기준을 도출하고자 한다.

이 논문의 구성은 다음과 같다. 2장에서 휴대폰 기본 모바일 지도 서비스 시스템 및 객체 지향 공간 데이터 모델을 소개하고, 3장에서는 시스템의 지도 제어 결과 및 지도

제어를 위한 객체 재사용 알고리즘을 소개한다. 4장에서 다양한 실험 결과를 살펴보고, 캐시 데이터의 재사용과 원시 데이터의 재전송 방법의 장단점을 살펴본다. 그리고 5장에서 결론을 맺는다.

2. 모바일 벡터 지도 서비스 시스템

2.1 시스템 구조 및 지도 검색 처리 과정

본 논문에서 적용하는 시스템 구조는 (그림 1)과 같이 크게 두 모듈로 나뉜다. 첫째, 지도 데이터베이스로부터 벡터 데이터를 수집하여 클라이언트로 전송하는 모바일 벡터 서버 모듈, 둘째, 서버로부터 전송받은 벡터 데이터를 휴대폰 단말기의 출력 창에 지도를 출력하는 모바일 벡터 클라이언트 모듈이다. 모바일 벡터 서버 모듈은 기존 GIS 서버[12]와 소켓 통신을 수행하며, 모바일 벡터 클라이언트 모듈과는 무선 소켓통신을 수행한다.

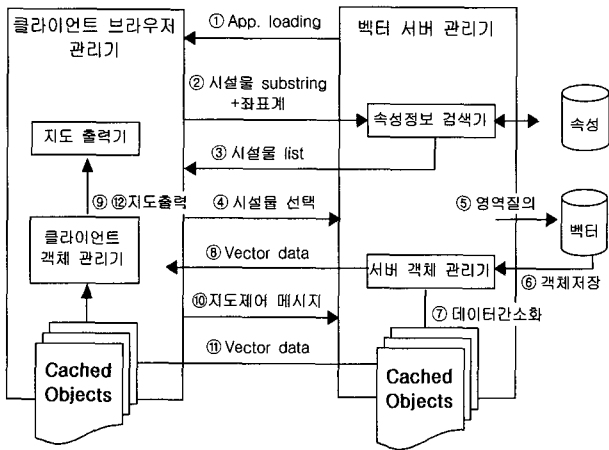
모바일 벡터 서버 모듈은 시설물 위치 데이터베이스로부터 시설물을 검색하는 속성정보 검색기, 모바일 환경에 적합하도록 벡터 데이터의 양을 축소시키는 Generalizator, Filter[13], 지도 데이터베이스로부터 수집한 벡터 데이터를 임시 저장하고 재사용하기 위한 서버 객체 관리기, 클라이언트와 통신을 처리하는 서버 메시지 처리기 등으로 구성된다.

모바일 벡터 클라이언트 모듈은 서버로부터 전송 받은 벡터 데이터를 관리하는 클라이언트 객체 관리기, 휴대폰 단말 출력 창에 맞게 지도를 그리는 지도 출력기, 서버와의 통신을 처리하는 클라이언트 메시지 처리기 등으로 구성된다. 시스템의 구현 환경은 다음과 같다.

- 모바일 벡터 서버 호스트 : Compaq Alpha Server DS10
- 모바일 벡터 서버 구현 : JAVA2 SE, JDBC
- 지도 DB : Cybermap Server Ver 2.0
- 속성 데이터베이스 : MySql
- 클라이언트 무선 단말기 : SK-VM Phone Emulator V.1.1
- 모바일 벡터 클라이언트 구현 : JAVA2 ME

모바일 벡터 서버 모듈과 클라이언트 모듈의 시스템 처리 메커니즘은 (그림 1)과 같다. 처음, 무선 단말기에 응용 프로그램이 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 볼륨을 축소시킨다(⑦). 이 결과 간소

화된 벡터 데이터를 클라이언트로 전송한다(⑧). 이후 클라이언트는 사용자의 요구에 따라 지도 제어 메시지를 서버로 전송[10]하여 필요한 데이터를 추가로 전송 받는다(⑩, ⑪). 이 때 유의할 것은 벡터 서버 관리기와 클라이언트 브라우저 관리기가 동시에 캐싱된 객체를 유지하고 있다는 것이다[1, 3].



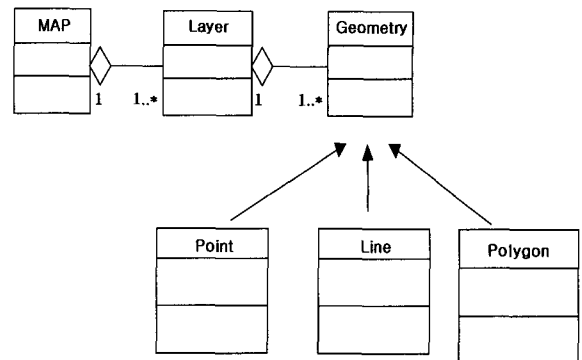
(그림 1) 처리 메카니즘

2.2 데이터 구조

서버 객체 관리기와 질의한 벡터 데이터를 서버 캐쉬에, 클라이언트 객체 관리기는 서버로부터 전송받은 벡터 데이터를 클라이언트 캐쉬에 각각 객체로 저장한다. 벡터 데이터의 임시 저장 객체 모델은 (그림 2)와 같다. (그림 2)에서 지도는 여러 개의 레이어로 구성되며 각 레이어는 여러 개의 Geometry로 구성된다. 그리고 Geometry는 점, 선, 면의 세가지 유형을 갖는다. 단, 하나의 레이어는 한 유형의 Geometry만 갖는다. 이 논문에서는 서버로부터 객체들을 전송 받을 때 이와 같은 객체 모델로 구조화한다. 이와 같이 객체를 구조화하여 캐싱하는 것은 다음과 같은 장단점이 있다.

장점으로는 구조화된 객체를 이용함으로써 효율적인 디스플레이 및 객체의 2차 가공이 가능하다는 점이다. 일반적으로 디스플레이 모듈은 레이어 및 객체 출력의 순서 조정으로 중첩표현을 막고 인식력을 높인다. 그러나 만약 객체를 구조화하지 않고 수신되는 즉시 출력한다면, 디스플레이의 품질은 낮아진다. 또 다른 장점으로는 구조화된 객체를 이용하며 클라이언트 위주의 단순 검색을 효율적으로 지원할 수 있다는 점이다. 다수의 클라이언트가 존재하는 환경에서 점 질의와 같은 단순 검색은 클라이언트 내에서 처리함으로써 서버의 부하를 줄일 수 있다.

반면, 단점은 객체 구조화에 많은 시간이 소요된다는 점이다. 이는 구조화를 위해 인스턴트 집합의 효율적인 관리가 필요하며, 객체의 삽입 및 삭제를 위해 메모리 압축 등 많은 추가 연산이 요구되기 때문이다.



(그림 2) 공간 객체 모델

(그림 3)은 서버와 클라이언트간의 메시지 전송을 위해 본 시스템에서 정의한 메시지의 구조이다. 하나의 맵을 전송하기 위해서는 레이어의 개수를 제일 먼저 보내고, 각 레이어의 타입에 따라 포인트(11), 라인(22), 폴리곤(33)의 정보를 보내고 가변 길이의 정보를 표현하기 위해 포인터 개수를 포함한 메시지를 전송한다.

Layer 개수	LayerType	Point 개수	시설물명 길이	시설물명	X좌표	Y좌표	y-1개의 순서쌍
x (short)	l1 (short)	y (short)	y (short)	(y bytes)	(byte)	(byte)	(길이,명,X,Y)

(a) 포인트 타입 메시지

Layer Type	Line 개수	Point 개수	X 좌표	Y 좌표	z1-1개의 순서쌍
22 (short)	y (short)	z1 (short)	(byte)	(byte)	(X,Y)

Point 개수	X 좌표	Y 좌표	z2-1개의 순서쌍
z2 (short)	(byte)	(byte)	(X,Y)

y-2개의 Line

(b) 라인 타입 메시지

LayerType	Polygon 개수	Point 개수	X 좌표	Y 좌표	z1-1개의 순서쌍
33 (short)	y (short)	z1 (short)	(byte)	(byte)	(X,Y)

Point 개수	X 좌표	Y 좌표	z2-1개의 순서쌍
z2 (short)	(byte)	(byte)	(X,Y)

y-2개의 Polygon

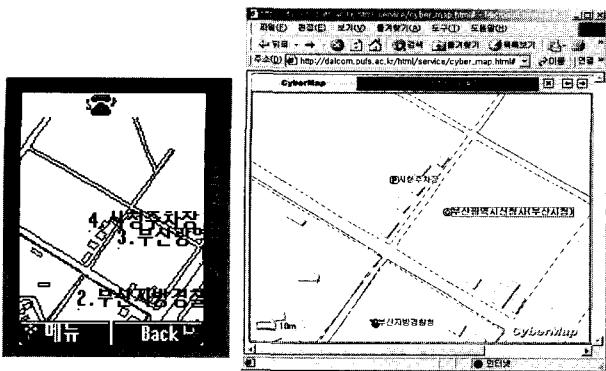
(c) 폴리곤 타입 메시지

(그림 3) 데이터 전송 메시지 구조

3. 효과적인 지도 제어를 위한 객체 재사용 기법

3.1 지도 제어 처리 결과

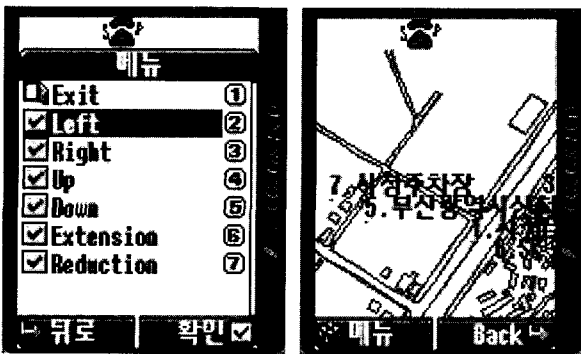
(그림 4)(a)는 클라이언트에서 시청을 입력한 후 부산 시청을 선택한 후 서버로부터 전송 받은 벡터 지도를 화면에 출력한 예이다. (그림 4)(b)는 유선 Web GIS 시스템[12, 14]에서 검색한 지도를 보이고 있다. 두 그림은 해상도의 차이일 뿐 내용상의 차이는 없음을 알 수 있다.



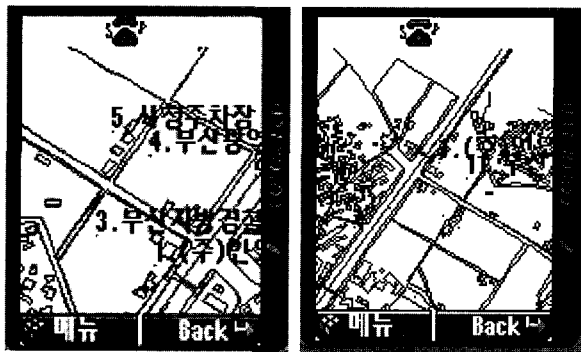
(a) 모바일 지도 서비스 (b) 유선 지도 서비스

(그림 4) 모바일 및 유선 지도 서비스의 출력 비교

(그림 5)(a)는 좌, 우, 상, 하의 스크롤과 확대 및 축소와 같은 지도 제어 명령 클라이언트 화면이다. (그림 5)(b),(그림 5)(c),(그림 5)(d)는 우 스크롤, 하 스크롤, 축소 명령을 수행한 예이다. 이때 각각 50%씩 이동하게 된다.



(a) 지도 제어 창 (b) Right Scroll



(c) Down Scroll (d) ZoomOut

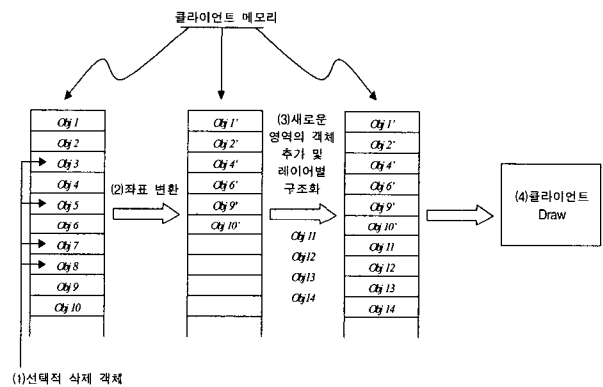
(그림 5) 지도 제어 처리 결과

3.2 지도 제어 알고리즘

지도 제어 명령이 발생했을 경우 모바일 클라이언트가 기존의 캐싱된 객체를 재사용하는 방법과 새로운 질의 영역의 모든 객체를 서버로부터 재전송 받는 방법이 있다. 일반적으로 객체 재사용 방법이 검색에 좋은 성능을 보일 것 같지만 장단점이 있다. 장점으로는 전송한 바와 같이 효과

적인 디스플레이 및 2차 가공의 용이성, 클라이언트 내에서 단순 질의 수행으로 인한 서버의 부하 감축, 재사용으로 인한 무선 데이터 전송의 최소화 등이 있다. 반면, 단점으로는 실제로 캐싱된 데이터의 재사용을 위해서는 클라이언트에서 기존 데이터의 선택적인 삭제 및 메모리 압축, 좌표변환, 새로운 데이터의 수신 및 객체 구조화, 구조화된 객체를 이용하여 화면상에 그리기와 같은 4가지 복잡한 단계를 거쳐야 하므로 낮은 계산능력을 가진 휴대폰에서는 많은 시간이 소요된다. (그림 6)은 이와 같은 4단계를 보여준다.

클라이언트 모듈에서 스크롤, 축소, 확대와 같은 지도 제어를 수행했을 경우 제일 먼저 수행되는 선택적인 삭제 및 압축은 새롭게 이동된 영역 밖에 존재하는 객체를 캐쉬에서 우선 제거하고 압축하는 과정으로 휴대폰의 제한된 메모리 용량에서는 반드시 필요한 과정이다. 다음으로 좌표변환 과정은 바뀐 출력 영역에 맞게 현재 캐싱된 객체의 좌표 변환을 하는 것으로 낮은 계산 능력을 가진 휴대폰 내에서의 좌표 변환은 많은 시간이 요구된다. 데이터의 수신 및 객체 구조화 과정은 새로운 영역에 존재하는 객체를 서버로부터 전송 받고 현재의 구조화된 객체에 레이어 별로 추가하는 작업이다. 마지막으로 캐쉬에 구조화된 객체를 이용하여 모바일 클라이언트 화면에 그리는 작업이 요구된다.



(그림 6) 캐시 데이터의 관리

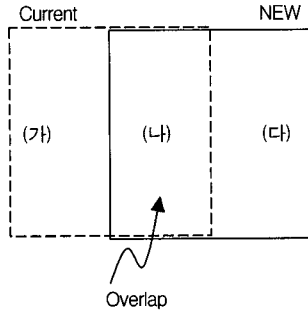
지도 제어는 크게 스크롤, 축소, 확대의 3명령으로 구분할 수 있다. 각각의 명령이 수행될 때 모바일 클라이언트에서 캐싱된 객체를 재사용하는 방법과 전송된 객체를 전혀 캐싱하지 않고 바로 출력하는 재전송 방법 또한 처리방법이 다르다.

3.2.1 스크롤 명령

(1) 재전송

- 서버 : 이미 출력된 객체와 상관없이 현재 (그림 7)과 같이 질의 영역으로부터 50% 이동한 새로운 질의 영역 (나,다)을 생성한 뒤 벡터데이터를 질의하여(그림 1) ⑤ 객체를 검색한 뒤 휴대장치의 좌표체계에 맞게 변환한 뒤 클라이언트로 전송.

- **클라이언트** : 새롭게 전송 받은 객체를 화면상에 출력. 새롭게 검색된 객체에 대한 캐싱을 하지 않음.



(그림 7) Right 스크롤

(2) 재사용

- **서버** : 이미 캐싱된 영역(그림 7)(나)은 처리대상에서 제외하고 새로운 영역(그림 7)(다)을 위한 질의 영역을 생성한 뒤 벡터 데이터를 검색한 뒤 클라이언트로 전송. 이 때 기존 캐싱된 객체중 약 절반정도는 좌표변환만을 수행하고, 나머지 절반은 새로운 질의로 검색된 객체가 구조화되면서 캐싱됨.
- **클라이언트** : (그림 7)(가) 영역에 해당하는 객체의 삭제 및 메모리 압축을 수행한 뒤 기존에 캐싱된 객체중 절반은 새로운 출력 영역에 맞게 좌표변환을 수행해야 하며, 나머지는 서버로부터 전송받아 캐시에 객체 구조화한 뒤 모바일 클라이언트 화면에 출력. 단, 출력 해상도가 변하지 않았기 때문에 레이어의 변화는 없음

3.2.2 확대 명령

(1) 재전송

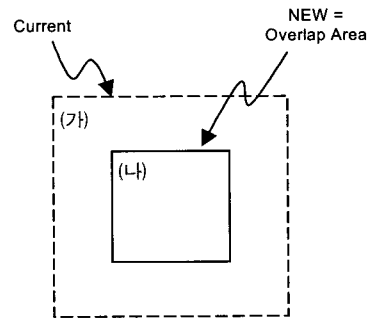
- **서버** : 서버는 현재 질의 영역의 중심점을 기준으로 상하좌우가 각각 50% 축소된 새로운 질의 영역을 생성한 뒤 벡터데이터를 질의하여 객체를 검색한 뒤 휴대장치의 좌표체계에 맞게 변환하고 벡터 데이터를 클라이언트로 전송. 확대명령으로 보다 자세한 지리 정보가 검색되어야 하므로 본 시스템에서는 <표 1>과 같이 8단계로 자료전송 레이어를 조절함. 즉, 확대될수록 아래단계로 진행되어 출력되는 지역은 줄어들지만 보다 많은 레이어가 서버로부터 클라이언트로 전송됨. 최종적으로 8단계이면 모든 레이어가 출력됨.
- **클라이언트** : 새롭게 전송받은 객체를 화면상에 출력. 새롭게 전송받은 객체들은 (그림 2)에서 제시한 공간 객체 모델에 따라 각 레이어 별로 여러 개의 점, 선, 면 객체가 존재하며, 여러 개의 레이어가 모여 하나의 지도(map)를 구성한다. 그리고 레이블(label) 정보로서 문자 정보 또한 수신된다.

<표 1> 확대/축소 단계

1 단계	전 국	최소레이어
2 단계	시 도	
3 단계	구 군	
4 단계	동 변	
5 단계	고 올	
6 단계	마 을	
7 단계	동 네	
8 단계	가 옥	전체레이어

(2) 재사용

- **서버** : 서버는 중첩된 영역(그림 8)(나)에 대해서 새로운 질의 영역을 설정하고, 확대로 인해 새롭게 추가되는 레이어에 대해서만 질의를 수행하여 검색한 뒤 클라이언트로 전송. 즉, 확대로 인해 중첩된 영역은 기존 영역 내에 존재하므로 서버에서는 현재 레이어들에 관한 정보는 추가 전송이 생략되며, 확대로 인해 요구되는 새로운 레이어는 해당 질의 영역 내의 모든 원소를 클라이언트로 전송함.
- **클라이언트** : 현재 캐싱된 객체중 중첩영역 내에 존재하는 객체는 좌표변환을 수행하며, 영역 외부에 존재하는 객체는 삭제한 후 압축을 수행함. 그리고 새로운 레이어의 객체들을 서버로부터 전송받아 캐싱함. 이 때 새로운 레이어에 대해서만 구조화를 수행하고 기존 레이어에 대한 추가 수신이 없으므로 객체의 재사용률이 매우 높음.



(그림 8) 확대

3.2.3 축소 명령

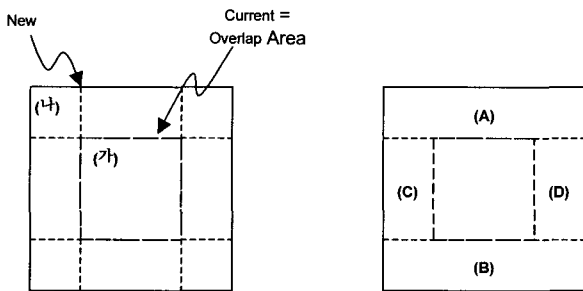
(1) 재전송

- **서버** : 확대 명령과 반대로 서버는 현재 질의 영역의 중심점을 기준으로 상하좌우가 각각 50% 확대된 새로운 질의 영역을 생성한 뒤 벡터 데이터를 질의하여 객체를 검색한 뒤 휴대장치의 좌표체계에 맞게 변환하고 벡터 데이터를 클라이언트로 전송. 축소 명령으로 전송되는 레이어 개수는 줄어들. 이는 과도하게 많은 데이터가 화면상에 출력되어 인식 불가능한 상태가 발생하는 것을 방지함.

- 클라이언트 : 새롭게 전송 받은 객체를 화면상에 출력. 일반적으로 축소 명령에 의해 보다 많은 수의 객체를 수신함.

(2) 재사용

- 서버 : 서버는 중첩된 영역(그림 9)(가)에 대해서는 새로운 데이터를 전송할 필요가 없고, 중첩 영역을 제외한 나머지 지역(그림 9)(나)에 대해서는 새로운 질의 영역을 설정하고, 해당 레이어를 축소 설정하여 검색한 뒤 클라이언트로 전송.
- 클라이언트 : 축소로 인해 중첩된 영역은 새로 출력될 영역의 일부이다. 따라서 캐싱된 객체중 축소해도 출력되어야 하는 레이어의 객체는 좌표 변환을 수행하고, 나머지 레이어의 객체는 삭제한 뒤 압축 수행. 그리고 나머지 새롭게 그려질 영역의 데이터는 서버로부터 전송받아 캐싱함. 축소는 기존 캐싱된 많은 영역과 레이어가 제외되어 객체 이용률이 떨어지고, 새롭게 질의되는 영역 또한 (그림 9)의 A부터 D까지 4번의 질의가 수행되며 넓은 지역의 많은 객체를 전송 받아야 하기 때문에 많은 무선 네트워크 전송 비용과 구조화 비용이 소요됨.



(그림 9) 축소

4. 성능 평가

전술한 바와 같이 캐싱된 객체의 재사용 방법은 많은 장점이 있지만 지도 제어 명령의 수행 시간의 지연이 발생하지 않는다는 전체 하에 적용되어야 한다. 전체 데이터의 재전송은 비교적 간단한 처리 알고리즘이지만 무선 대역폭의 사용량이 증가하는 반면, 캐싱된 객체의 재사용은 삭제, 압축, 삽입 연산이 각 명령마다 복잡하게 처리되어야 하는 단점이 있다. 또한 객체 재사용 방법은 지도 제어 연산에 따라 복잡도가 조금씩 다르다. 또한 낮은 계산 능력, 무선 통신과 같은 모바일 휴대폰 환경에서는 재전송과 재사용 방법의 수행시간에 대한 실험을 통해 성능이 평가되어야 한다.

실험 평가를 위해 2장에서 제시한 시스템을 이용하였으며, 전체 1500회의 수행 시간 테스트를 진행하였고, 실험 대상 지역은 객체의 개수가 최소 12개로부터 최대 1500여

개에 이르는 다양한 곳을 선택하였다. 그리고 좌우상하 스크롤, 축소, 확대 명령을 같은 비율로 수행하였다.

4.1 전체 수행 결과

<표 2>는 전체 수행 결과를 종합적으로 정리한 것이다. 실험 결과 평균 수행 시간은 캐시를 이용한 재사용 방법이 좋은 성능을 보임을 알 수 있으며, 객체 이용률을 약 50%이다.

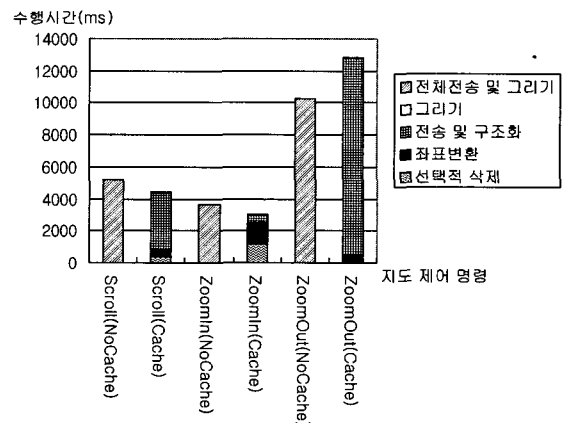
<표 2> 전체 수행 결과

	제어 명령 평균 수행 시간	객체 재사용률
재전송(NoCache)	5626.058	0
재사용(Cache)	5281.104	50.08%

4.2 명령별 수행 결과

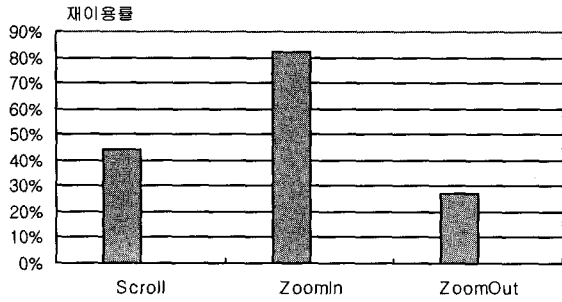
(그림 10)은 각 명령과 데이터 이용 방법에 따른 전체 수행 시간 및 세부과정의 수행 시간을 도식화한 것이다. 각 명령별 전체 수행시간을 보면 스크롤과 확대 명령에서는 캐시를 이용하는 것이 좋은 성능을 보인 반면, 축소 명령에 대해서는 캐싱된 데이터를 재사용하는 방법이 좋지 않은 성능을 보였다.

그리고 세부 과정의 수행 시간을 살펴보면 확대 시에는 선택적 삭제와 좌표 변환에 상대적으로 많은 시간이 소요되고, 축소시에는 새로운 질의 영역에 대한 데이터 전송 및 구조화에 많은 시간이 소요된다. 이는 무선 모바일 클라이언트 환경의 특성을 그대로 보여 주고 있음을 알 수 있다.



(그림 10) 명령별 수행 시간

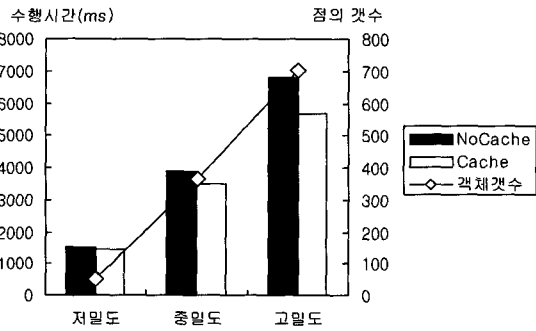
(그림 11)은 캐싱된 객체를 이용하는 방법을 적용한 각 지도 제어 명령별 객체 재사용률을 나타낸 것이다. (그림 11)에서 확대는 80% 이상의 좋은 이용률을 보이며, 축소는 재사용률이 매우 떨어졌다. 이와 같은 재사용률이 결국 수행시간에도 영향을 미쳐 축소시에는 캐시 데이터 재사용 방법이 재전송 방법보다 성능이 좋지 않게 된다.



(그림 11) 명령별 객체 재사용률

4.3 지역 밀도별 수행 결과

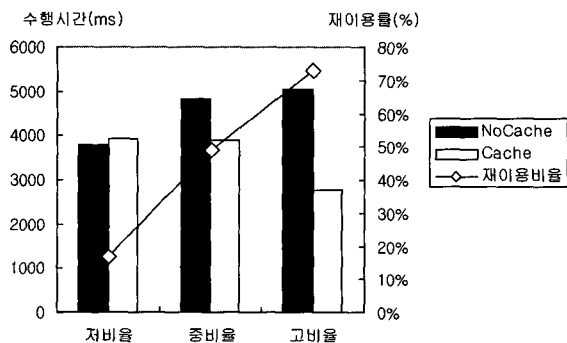
(그림 12)는 지역 밀도별로 스크롤 명령을 수행한 결과로서 화면상의 출력되는 객체의 개수가 많으면 많을수록 객체 캐싱 방법이 데이터 재전송 방법보다 더 좋은 성능을 보임을 알 수 있다. (그림 12)에서 저밀도는 화면상에 출력된 점의 갯수가 50개 정도이고, 고밀도는 평균 700개 정도이다.



(그림 12) 지역 밀도별 스크롤 명령 수행 결과

4.4 객체 재사용률별 수행 결과

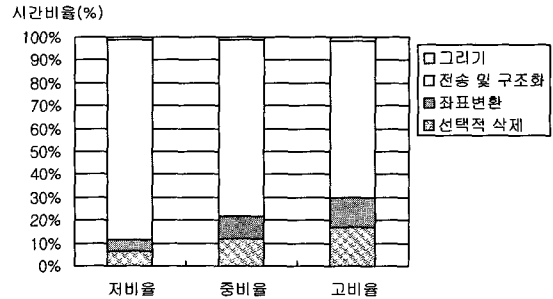
(그림 13)은 캐싱된 객체의 재사용률에 따른 스크롤 명령의 수행 시간을 나타낸 것으로 객체의 재사용률이 높을수록 캐시를 이용한 방법이 더 좋은 성능을 보인다.



(그림 13) 객체 재사용 비율과 수행 시간

(그림 14)는 객체의 재사용률에 따른 캐시 이용 방법에서 세부 수행 항목의 상대적인 수행 시간 비율을 나타낸 것으로

서 재사용률이 높을수록 새로운 질의 영역에 대한 데이터의 전송 및 구조화 비용의 비중이 점차 낮아짐을 알 수 있다.



(그림 14) 객체 재사용 비율과 세부 수행 시간

이상과 같은 실험결과를 종합해보면 다음과 같다. 객체의 재사용 방법이 스크롤과 확대 명령에는 좋은 성능을 보이지만, 축소 명령에는 좋지 않은 성능을 보인다. 따라서 축소 명령에서는 캐싱 데이터를 이용하지 않는 방법이 더 좋을 것으로 판단된다. 그리고 고밀도 지역에서 객체 재사용률이 높을수록 더 좋은 성능을 보였다. 이런 결과는 모바일 폰을 기반으로 하는 무선 벡터 지도 서비스 환경에서도 객체의 캐싱에 의한 재사용 방법이 충분히 적용될 수 있음을 시사하는 것이다.

이런 실험 결과를 토대로 도출된 스크롤, 축소, 확대와 같은 지도 제어 연산별 객체 재사용 기법은 (그림 15)와 같다. (그림 15)에서 축소 명령에 대해서는 기존의 캐싱 데이터를 이용하진 않지만 추가 사용을 위해 구조화를 수행하는 기법을 택하였다.

```

CASE : SCROLL
layer ← unchange
FOR ( all Obj cache )
  IF( Obj ∩ QR[new] = ∅ )
    delete Obj from cache
  ELSE conversion coordinates of Obj
cache memory compaction
Objects = Query(QR(new) - QR(current), layer)
append Objects into cache
CASE : EXTENSION
layer ← stepwise increment
FOR ( all Obj cache )
  IF( Obj ∩ QR[new] = ∅ )
    delete Obj from cache
  ELSE conversion coordinates of Obj
cache memory compaction
Objects = Query(QR(new), layer-old_layer)
append Objects into cache
CASE : REDUCTION
layer ← stepwise decrement
FOR ( all Obj cache )
  delete Obj from cache
Objects = Query (QR(new), layer)
append Objects into cache
    
```

(그림 15) 지도제어 의사 코드

5. 결 론

이 논문은 기존 PC 기반 클라이언트 환경에서는 자연스럽게 적용되고 있는 객체 캐싱 방법이 모바일 휴대폰 기반 무선 지도 서비스에서도 사용할 수 있는 가를 검증하기 위해 연구하였다. 왜냐하면 모바일 지도 서비스는 휴대폰의 적은 메모리 용량, 낮은 계산 능력, 무선 통신, 불특정 다수가 동시에 접근하는 서버의 환경이 매우 특수하기 때문이다. 또한 지금까지 제공되지 않았던 지리 객체의 처리 기능을 가진 시스템을 J2ME를 기반으로 하였기 때문이다.

캐싱된 객체의 재사용을 위해서는 클라이언트에서 기존 데이터의 선택적인 삭제, 좌표변환, 메모리 압축, 새로운 데이터의 수신 및 객체 구조화 과정이 요구된다. 따라서 휴대폰을 이용한 지도 제어시 캐싱된 객체의 재사용 방법의 성능은 다양한 실험을 통해 수행 시간을 기준으로 평가할 수 밖에 없었다.

실제 데이터를 제공하는 서버를 이용한 실험 수행 결과 객체의 재사용 방법이 대체로 좋은 성능을 보였다. 그리고 고밀도 지역에서 객체 재사용률이 높을수록 더 좋은 성능을 보였다. 그리고 이런 실험 결과를 토대로 도출된 객체 재사용 방법은 무선 벡터 지도 서비스 환경에서도 객체의 캐싱에 의한 재사용 방법이 충분히 적용될 수 있을 것으로 판단된다.

향후 연구과제로는 실험 결과로 도출된 각 세부 항목의 수행 시간을 줄일 수 있는 방안으로서 데이터의 전송량을 최대한 줄이고 구조화 시간을 단축하는 연구를 수행하고자 한다.

참 고 문 헌

[1] P. Cao, C. Liu, "aintaining Strong Cache Consistency in the World-Wide Web," IEEE Transactions on Computers, Vol.47, No.4, pp.445-457, 1998.
 [2] J. Jing, A. Elmagarmid, "Client-Server Computing in Mobile Environments," ACM Computing Surveys, Vol.31, No.2, 1999.
 [3] A.Kahol, S.Khurana, S. Gupta, P.Srimani, "An Efficient Cache Maintenance Scheme for Mobile Environment," International Conference on Distributed Computing Systems, 2000.
 [4] R. Laurini, D. Thompson, "Fundamentals of Spatial Information Systems," Academic Press, 1992.
 [5] 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.

[6] W. Peng, M. Chen, "A Dynamic and Adaptive Cache Retrieval Scheme for Mobile Computing Systems," Int. Conf. of Coopis98, pp.251-258, 1998.
 [7] K. Stathatos, N. Roussopoulos, J.Baras, "Disseminating Updates to Mobile Clients," Technical Research Report in the CSHCN at University of Maryland, 1998.
 [8] M. Wong, W. Leung, "A Caching Policy to Support Read-only Transactions in a Mobile Computing Environment," Technical Report CS-TR-95-07, Dept. of Computer Science, Chinese University of Hong Kong, 1995.
 [9] J. Yuen, E. Chan, K. Lam, H. Leung, "Cache Invalidation Scheme for Mobile Computing Systems with Real-time Data," ACM SIGMOD Record, Vol.29, No.4, 2000.
 [10] W. Abdelsalam, "Maintaining Quality of Service for Adaptive Mobile Map Clients," Master thesis of Mathematics in Computer Science in University of Waterloo, Canada, 2001.
 [11] <http://java.sun.com/j2me/>.
 [12] <http://www.cybermap.co.kr/cm2000/newhome/index.html>.
 [13] 김미란, 최진오, "Generalization과 filtering을 이용한 무선지도 터베이스의 동적 생성 기법", 정보처리학회논문지, 제8권 제4호, 2001.
 [14] 김진덕, 최진오, "휴대폰 기반 벡터 지도 서비스를 위한 클라이언트-서버 시스템의 캐싱기법", 한국해양정보통신학회 논문지, 제6권 제3호, 2002.



김진덕

e-mail : jdk@dongeui.ac.kr
 1993년 부산대학교 컴퓨터공학과(공학사)
 1995년 부산대학교 컴퓨터공학과(공학석사)
 2000년 부산대학교 컴퓨터공학과(공학박사)
 1998~2001 부산정보대학 정보통신계열
 전임강사

2001~현재 동의대학교 컴퓨터공학과 조교수
 관심분야 : 데이터베이스, 공간DB, 공간색인, 모바일 데이터베이스



최진오

e-mail : jochoi@taejo.pufs.ac.kr
 1991년 부산대학교 컴퓨터공학과(공학사)
 1995년 부산대학교 컴퓨터공학과(공학석사)
 2000년 부산대학교 컴퓨터공학과(공학박사)
 1998~2000 경동대학교 컴퓨터공학과 전임
 강사

2000~현재 부산외국어대학교 컴퓨터공학과 조교수
 관심분야 : 데이터베이스, GIS, 분산DB, 트랜잭션, 모바일 데이터베이스