모바일 맵 서비스를 위한 벡터와 래스터 기법의 비교 평가

최진오*

Comparison and Evaluation of Vector and Raster Methods for Mobile Map Services

Jin-Oh Choi*

요 약

모바일 GIS를 구현하는 방식은 휴대 단말기로 전송되는 지도 데이터의 형식에 따라 벡터(Vector) 기법과 래스터(Raster) 기법으로 나뉜다. 각 기법에 따라 모바일 지도 서비스를 위한 서버 및 클라이언트 프로그램의 구현 방법이 달라지며 각각 서로 다른 측면에서 장단점을 지니고 있다. 본 논문은 이 두 접근 방법을 구현하고 실험을 통해 그 장단점을 비교한다. 전송 성능, 지도 품질 등 다양한 요소를 비교하고 평가한다.

ABSTRACT

There are two approaches to construct mobile GIS, Vector and Raster methods, according to the map data transformation format from server to mobile client. Each method requires a different implementation architecture of server and client modules for mobile map services. And each have advantages and disadvantages at the different aspects. This thesis implements these two approaches, thus, compares the each merits, by experiments. They include the transmission performance, map quality, and so on.

키워드

mobile vector GIS, mobile raster GIS, mobile phone

Ⅰ. 서 론

모바일 전자 지도 서비스는 기존의 유선 인터넷 환경과 같이 래스터(raster) 방식과 벡터(vector) 방식으로 나뉜다. 래스터 방식의 서비스는, 서버에서 검색한 전자지도를 이미지로 변환하고 이것을 WML(Wireless Markup Language)[1] 문서에 포함시켜 클라이언트로 전송하거나 소켓(socket)을 통해 직접 전송하는 방식을 취한다. WML 문서에 포함시켜 이미지로 전송할 경우 클라이언트에는 WAP 브라우저(browser) 프로그램이 필요하다. 그리고 소켓을 이용한 전송의 경우 클라이언트에 이미지를 출력하는 전용 브라우저 프로그램이 필요하다. 전용 브라우저는 모바일 플랫폼에 따라 J2ME[2]나 C 언어로 개발해야 한다. 래스터 방식의 공통점은 서버에서 클라이언트로 이미지 지도를 전송한다는 것이다.

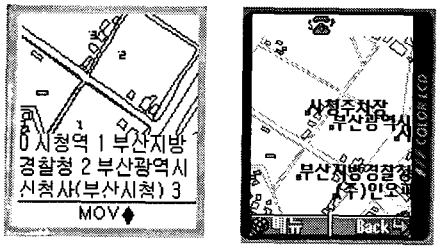
벡터 방식의 서비스는, 서버에서 검색한 지도 벡터 데이터를 그대로 소켓을 통해 클라이언트로 전송하고, 클라이언트에서 수신한 공간 데이터를 전용 브라우저 프로그램으로 지도를 직접 그리는 방식을 취한다.

그림 1은 각각의 지도 출력 예를 보인다. 그림 1 (가)는 래스터 지도를, (나)는 벡터 지도를 클라이언트에서 출력한 구현 화면이다. 그림 1 (가)는 클라이언트가 지도 이미

* 부산외국어대학교 컴퓨터공학부

접수일자 : 2006. 5. 23

지와 범례를 WAP 환경의 WML로 받아 출력한 화면이고, 그림 1 (나)는 소켓으로 받은 벡터 데이터를 J2ME 프로그램으로 클라이언트에서 직접 지도를 그린 출력 화면이다.



(가) 래스터지도 (나) 벡터지도

그림 1. 두 기법의 지도 출력 예

Fig. 1. Map Display Examples of two approaches

래스터와 벡터 방식의 지도 출력 기법의 차이점을 비교해보면 다음과 같다. 첫째, 서버에 걸리는 부하 차이이다. 래스터 방식은 서버에서 모든 처리가 이루어져야 하며 클라이언트에서 출력될 최종 지도 화면을 이미지로 만들어야 한다. 반면, 벡터 방식에서는, 서버는 지도 데이터를 벡터 형식으로 클라이언트로 전송만 하면 된다. 따라서 서버의 부하는 래스터 방식에서 더 많이 걸리게 된다.

둘째, 클라이언트에서의 부하 차이이다. 래스터 방식은 서버에서 전송되는 데이터를 그대로 이미지 출력만 하면 되지만, 벡터 방식은 벡터 데이터를 일일이 클라이언트 캔버스(canvas) 화면에 그리는 작업을 수행하여야 한다. 심지어, 래스터 방식에서 WAP 환경을 사용할 경우 모바일 지도 서비스를 위한 별도의 클라이언트 모듈이 필요없다.

셋째, 클라이언트에서 지도를 출력하는 융통성에 차이가 있다. 래스터 방식은 클라이언트 디바이스(device)의 특성을 고려한 지도 출력이 어렵고 표준화된 형식과 크기의 이미지를 생성해야 하는 제약이 따른다. 벡터 방식은 클라이언트에서 수신 받은 벡터 데이터를 디바이스 특성에 맞게 변환 출력하거나 필요에 따라 다양한 방식으로 출력할 수 있는 유연성이 있다.

넷째, 서버에서 클라이언트로 전송되는 데이터의 크기에 차이가 있다. 래스터 방식은 이미지를 전송하기 때문에 거의 일정한 데이터 볼륨을 전송하게 된다(특히 BMP 포맷의 이미지일 경우). 반면 벡터 방식은 지도에 포함된 포인트 개수에 따라 상이한 벡터 데이터 볼륨을 제공한다. 따라서 서버의 응답시간이 래스터 방식은 거의 일정한 반면 벡터 방식은 가변적일 수 있다.

다섯째, 클라이언트에서의 지도 활용 가능성 정도가 서로 다르다. 래스터 방식은 이미지를 전송받기 때문에 클라이언트에서 공간분석(spatial analysis), 캐싱(caching) 등의 목적으로 지도 데이터를 응용할 가능성이 거의 없다. 반면, 벡터 방식은 서버로부터 전송받은 벡터 데이터를 저장해 두고 네트워크 분석이나 필요에 따른 지도 변형, 캐싱 등의 목적으로 활용 및 재사용이 가능하다.

본 논문에서는 이러한 두 기법의 상이한 특성을 정확히 파악하기 위해 각각의 모바일 지도 서비스 시스템을 설계 구현하고 다양한 방법으로 비교 평가하기 위한 실험을 전개한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 설명하고, 3장에서 두 접근 방법의 시스템 구현 설계 개요를 소개하며, 4장에서 이들 두 기법의 장단점을 실험하고 평가한다. 결론은 5장에서 맺는다.

Ⅱ. 관련 연구

모바일 지도 서비스 구현을 위한 플랫폼은 WAP 기반 방식, KVM(K Virtual Machine) 기반 방식[3], BREW (binary runtime environment for wireless) 기반 방식[4], WIPI(Wireless Internet Platform for Interoperability) 기반 방식[5] 등이 있다. 래스터 지도 서비스 구현은 모든 플랫폼에서 구현 가능하지만 벡터 지도 서비스는 WAP 기반 방식으로는 지원이 불가능하다.

무선 인터넷을 통한 국내 상용 전자지도 서비스 동향은 대부분이 성능 문제 때문에 래스터 지도를 선택하고 있다. 일부 업체에서 벡터 방식의 무선 지도 서비스[6]를 시행하고 있으나 대부분 모바일 서비스 전용의 간소화된 GIS 데이터베이스에 의존하고 있는 실정이다. 국외의 경우도 벡터 방식의 모바일 지도 서비스는 일반화되지 못하고 있다[7][8].

OGC(Open GIS Consortium)에서 제정한 웹 지도 서버 관련 규격[9]으로 WMS(Web Map Server), WFS(Web Feature Server), 그리고 WCS(Web Coverage Server)가 있다. WMS는 래스터 방식의 지도 처리를 위한 규격이며 WFS는 벡터 방식의 지도 처리를 위한 규격이다. OGC의 관련 규격들은 각각의 지도 처리 방법에 따라 클라이언트에서 서버로 데이터를 요청하는 API를 제시하고 있다.

본 논문에서 비교 평가하고자 하는 두 접근 방법은 OGC의 표준에 따라 구현하지는 않고 별도의 인터페이스를 사용한다. 본 논문에서는 두 기법의 장단점을 실험을 통해 비교 분석하고, 이를 바탕으로 벡터 방식의 지도 서비스에서 극복되어야 할 문제점을 고찰하고자 한다.

Ⅲ. 시스템 구현 설계

그림 2는 모바일 지도 서버의 구조이다. 이 그림에서 'Raster Data Processor'와 'Vector Data Processor'가 각각 래스터 지도와 벡터 지도를 생성하는 모듈이다.

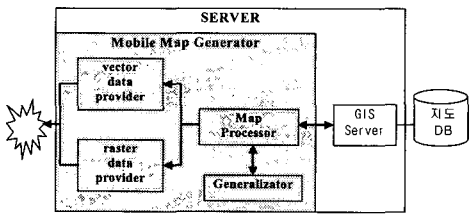


그림 2. 모바일 맵 서버
Fig. 2. Mobile Map Server

그림 2의 'Map Processor'는 GIS 서버로부터 검색한 지도를 'Generalizator'[13]를 통해 모바일 환경에서 출력 가능한 지도로 간소화한 후 래스터나 벡터 데이터 처리기로 제공한다.

그림 3은 'Raster Data Processor'를 보이고 있다. 'Raster Data Processor'는 제공받은 공간 데이터로부터 'Image Processor'를 통해 이미지 파일을 생성한다. 'Page Processor'는 이 이미지와 속성정보를 포함한 WML 문서를 생성한다.

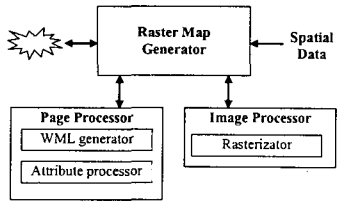


그림 3. Raster Data Processor 구조
Fig. 3. Structure of Raster Data Processor

그림 4는 'Vector Data Processor'의 구조를 보이고 있다. 이 구조는 'Raster Data Processor'보다 비교적 간단하다. 넘겨받은 공간 데이터를 약속된 형식으로 소켓을 통하여 전송하기만 하면 되기 때문이다. 'Server Cache Manager'는 이후 서버 캐싱을 위해 전송한 지도를 구조화시켜 저장해 둘 수 있도록 지원한다. 이는 벡터 방식에서 얻을 수 있는 장점 중 하나이다.

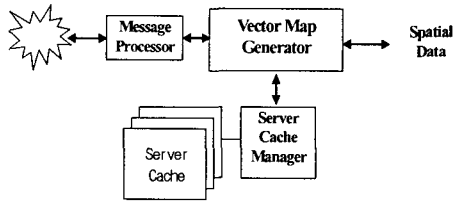


그림 4. Vector Data Processor 구조
Fig. 4. Structure of Vector Data Processor

서버 구조에 이어 클라이언트 구조는 다음과 같다. 먼저, WML을 이용하는 래스터 방식은 클라이언트에 WAP 브라우저를 필요하다. 소켓을 이용하는 래스터 방식은 클라이언트에 이미지를 재생하여 출력하는 모듈이 필요하다.

벡터 방식의 서버에 대한 클라이언트에는 전송받은 벡터데이터를 휴대 단말기에 직접 그리는 브라우저 모듈이 필요하다. 그림 5는 KVM 플랫폼에서의 클라이언트 모듈인 'Mobile Vector Map Browser'의 구조를 보이고 있다. 그림 5에서 브라우저는 소켓으로 전송받은 벡터 데이터를 메시지 처리를 통해 'Client Cache'에 저장한 후 또는 저장하지 않고 직접 'Map Drawer'로 캔버스에 출력한다. 필요에 따라 'U/I Processor'가 지도 판독도를 높이기 위한 보조 작업을 수행할 수 있다.

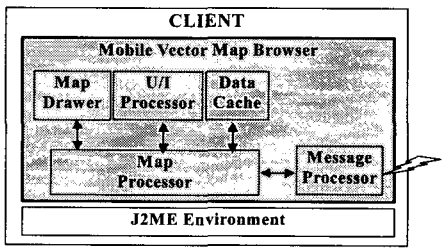


그림 5. 벡터 클라이언트 구조
Fig. 5. Structure of Vector Client

Ⅳ. 구현 실험

본 논문에서 설계 구현하고 실험한 환경은 다음과 같다.

표 1. 실험 환경
Table 1. Experiment Environments

Server	Compaq Alphaserver DS10
Map Database	Cybermap Server Ver 2.0
Raster Data Processor	Servlet Web Server
Vector Data Processor	JAVA 2 SE
Raster Client	NETPLE WAP Browser Phone Emulator Ver 1.1
Vector Client	SK-VM Phone Emulator Ver 1.1, J2ME
실제 휴대폰 테스트	삼성전자 SCH-X570

실험은 상용 지도인 'Cybermap'을 사용하였으며, 실험으로 도출된 수치는 500m 영역 내에 포인트 개수가 500개 단위로 500개에서 2000개까지 각각 이내인 100개 표본 영역을 선택하여 평균값을 구한 것이다.

그림 6은 'Vector Data Processor'와 'Raster Data Processor'의 서버에서의 처리 소요 시간을 측정하여 비교한 결과이다. 즉, 지도 데이터베이스로부터 지도를 검색하여 전송할 수 있는 상태까지 소요되는 처리 시간을 측정한 것이다. 실험결과, 서버 프로세서가 처리해야 할 데이터 크기는 두 기법이 큰 차이를 보이지 않지만 포인트 수가 증가할수록 래스터 방식에서 서버 처리 시간이 벡터 방식보다 커지는 결과를 보였다. 그 이유는 벡터 방식에 비하여 래스터 방식은 서버에서 이미지를 그려야 하는 오버헤드가 더해지기 때문이다. 이것은 포인트 수가 많아질수록 증가할 것이다.

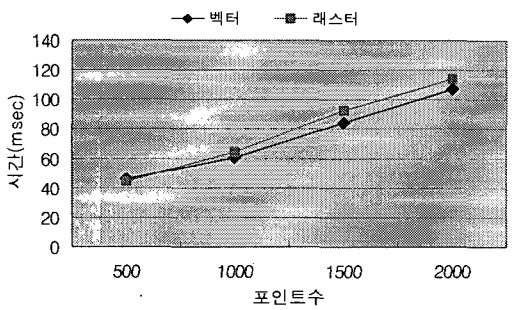


그림 6. 서버 처리시간 비교
Fig. 6. Comparison of Server Processing Time

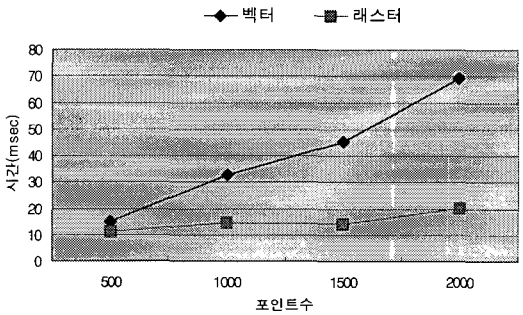
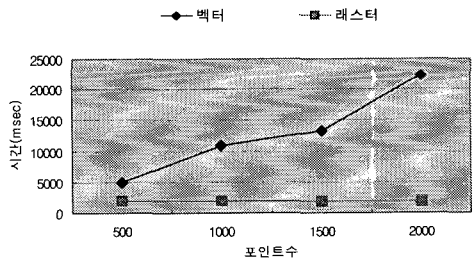
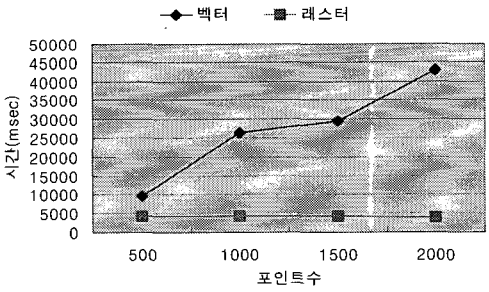


그림 7. 서버 전송시간 비교
Fig. 7. Comparison of Server Transfer Time

그림 7은 서버에서 클라이언트로 데이터를 전송 시작하여 완료하는데 소요되는 시간을 측정한 실험 결과이다. 즉, 서버에서의 순수 데이터 전송 소요 시간을 측정한 것이다. 벡터 방식은 포인트 수가 증가할수록 처리시간이 증가하지만 래스터 방식은 거의 일정함을 보였다. 다만 래스터 방식은 WML 문서를 이미지와 함께 전송하며 WML 문서의 크기는 범례가 많을수록 커진다.



(a) 에뮬레이터 실험
(a) Emulator Experiment



(b) 실제 휴대폰 실험
(b) Real Phone Experiment

그림 8. 클라이언트 처리시간 비교
Fig. 8. Comparison of Client Processing Time

그림 8은 클라이언트가 서버에서 데이터를 전송받기 시작하여 출력 완료하는데 소요되는 시간을 측정하고 비교한 결과이다. 실험 결과 값은 각 포인트 개수 근사치의 지도 영역 표본을 10회 반복 측정한 결과의 평균값이다. 실제 휴대폰 테스트 결과는 통신망의 여건에 따라 가변적이기는 하지만 에뮬레이터에서의 결과와 유사한 양상을 보였다. 여기서, 래스터 방식은 일정한 크기의 이미지를 전송받기 때문에 포인트 수에 관계없이 일정한 응답 속도를 보인다. 벡터 방식은 포인트 수가 증가함에 따라 클라이언트에서의 처리 시간이 증가함을 알 수 있다. 이는 단말기 출력 화면에 직접 지도를 그리는 작업 소요 시간 때문이다.

그림 9는 래스터와 벡터 방식이 각각 전송하는 데이터 크기를 측정 비교한 결과이다. 이 실험 결과에서 벡터 방식의 전송 데이터의 크기가 래스터 방식보다 작음을 알 수 있다. 또한 래스터 방식은 흑백 이미지나 컬러 이미지나 포인트 수에 관계없이 일정한 데이터 크기를 유지함을 알 수 있다.

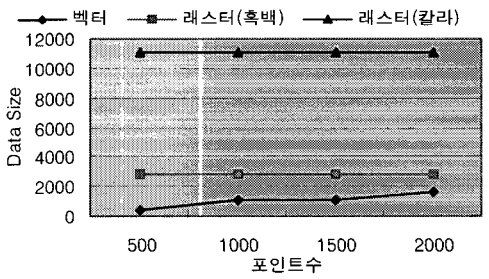


그림 9. 데이터 크기 비교
Fig. 9. Comparison of Data Size

V. 결 론

본 논문은 모바일 환경에서 지도 서비스를 지원하는 두 가지 기법, 래스터 기반, 벡터 기반 기법을 각각 설계 구현하고, 실험을 통하여 비교 분석한 결과를 보였다. 여기서 설계하여 보인 내용은 래스터 지도를 생성하고 클라이언트로 전송하는 'Raster Data Processor', 벡터 지도를 생성하고 클라이언트로 전송하는 'Vector Data Processor', 그리고 벡터 지도를 클라이언트에 출력하기위한 'Mobile Vector Map Browser'이다.

실험 결과는 래스터 방식과 벡터 방식이 각각 장단점이 있음을 보였다. 래스터 방식은 일정한 응답 속도를 보장하며, 자원이 부족한 클라이언트 휴대 단말기에 적합한 방법임을 알 수 있었다. 반면 벡터 방식은 클라이언트로 전송되는 데이터의 크기가 작기 때문에 전송 시간이 짧으며, 서버의 부하를 상대적으로 줄여 주는 장점이 있었다.

향후 본 논문의 연구 결과를 바탕으로 래스터 방식과 벡터 방식을 혼합 방법을 연구할 필요성이 있다. 즉, 각 방식의 장점만을 취하여 서로의 단점을 보완할 수 있는 기법을 고안할 수 있을 것이다. 이 혼합 기법은 벡터 방식의 느린 클라이언트 응답 속도 문제를 래스터 방식으로 극복하며, 래스터 방식의 단순함을 벡터 데이터로 보완할 수 있을 것으로 기대된다.

참고문헌

[1] WAP Forum Specifications, WAP June 2000 Conformance Release

[2] "Java Platform, Micro Edition (Java ME) Overview", http://java.sun.com/javame/overv iew.html

[3] White Paper, "The K virtual machine (KVM)", http://java.sun.com/products/cldc/ wp/index .html

[4] "About BREW", http://brew.qualcomm.com/ brew/en/ about/about_brew.html

[5] "WIPI 규격", http://wipi.or.kr

[6] http://www.pointi.com

[7] http://www.nttdocomo.com

[8] http://www.viamichelin.com

[9] "OpenGIS Specifications", http://www.open geospatial. org/specs/

[10] Y. Chiang, C. Knoblock, C. Chen, "Image and shape analysis - user interaction: Automatic extraction of road intersections from raster maps", Proceedings of the 13th annual ACM international workshop on GIS 2005

[11] B. Yang, "A multi-resolution model of vector map data for rapid transmission over the Internet", Computers & Geosciences, Volume 31, Issue 5, June 2005, Pages 569-578

[12] J. Casademont, E. Aguilera, J. Paradells, A. Rojas, A. Calveras, F. Barcelo, J. Cotrina, "Wireless technology applied to GIS", Computers & Geosciences, Volume 30, Issue 6, July 2004, Pages 671-682

[13] 최진오, "모바일 GIS를 위한 공간 데이터 간소화 기법에 대한 연구", 해양정보통신학회논문지, Vol 8, No 1, pp 150~157, 2004.02

저자소개

최 진 오(Jin-Oh Choi)

1991년 부산대학교 컴퓨터공학과 공학사
1995년 부산대학교 대학원 컴퓨터공학과 공학석사
2000년 부산대학교 대학원 컴퓨터공학과 공학박사
1998. 3 ~ 2000. 2 경동대학교 정보통신공학부 전임강사
2000. 3 ~ 현재 부산외국어대학교 컴퓨터공학부 부교수
※관심분야 : 공학데이터베이스, 지리정보시스템, 모바일 GIS