

모바일 GIS를 위한 벡터 데이터 경량화 기법

윤근정* · 김혜영** · 전철민***

The Optimization of Vector Data for Mobile GIS

Geunjung Youn* · Hyeyoung Kim** · Chulmin Jun***

요 약

기존의 유선 환경기반의 GIS 솔루션으로 무선 환경에서 서비스를 하기에는 제한된 통신 속도, 처리 속도, 화면 사이즈 등의 한계점이 존재한다. 본 연구에서는 이러한 제한된 특성들을 극복하기 위하여 모바일 환경에서의 벡터 데이터의 경량화 기법에 대하여 네 가지 단계로 제시하였다. 본 논문에서 제시한 방법을 검증하기 위하여 강남구 지역을 대상으로 실험을 실시하였다. 기존의 유선 환경에서 구축되어진 GIS 엔진과 본 논문에서 제시한 방법을 적용한 경량화된 GIS 엔진을 동일한 환경에서 성능 평가를 실시하였고 이에 따라 응답 데이터 크기, 초당 요청 처리 개수, 평균 요청 처리 시간을 비교 분석하였다. 그 결과 경량화가 적용된 GIS 엔진이 기존의 엔진과 비교하여 상당한 성능 향상이 있었다는 것을 입증하였다.

주요어 : 모바일 GIS, 벡터데이터, 경량화

ABSTRACT : Providing services in a wireless environment with existing wired-based GIS solutions have many limitations such as slow communication, processing rates and screen size. This study suggested data optimization techniques in a mobile environment to overcome those limitations in four steps. In order to test the methods suggested in the study, experiments are conducted using Gangnam-gu as a test site. An existing GIS engine built in a wired environment was compared with the optimized GIS engine from the study in terms of performance in the same environment. They were also compared and analyzed in terms of response data size, number of requests processed per second, and average time to process a request. The results

*선도소프트 선도GIS연구소 과장(gjyoun@sundosoft.com)

**서울시립대학교 공간정보공학과 박사과정(mhw3n@uos.ac.kr)

***서울시립대학교 공간정보공학과 교수, 교신저자(cmjun@uos.ac.kr)

proved that the proposed engine shows significant improvements in performance compared with the wired GIS.

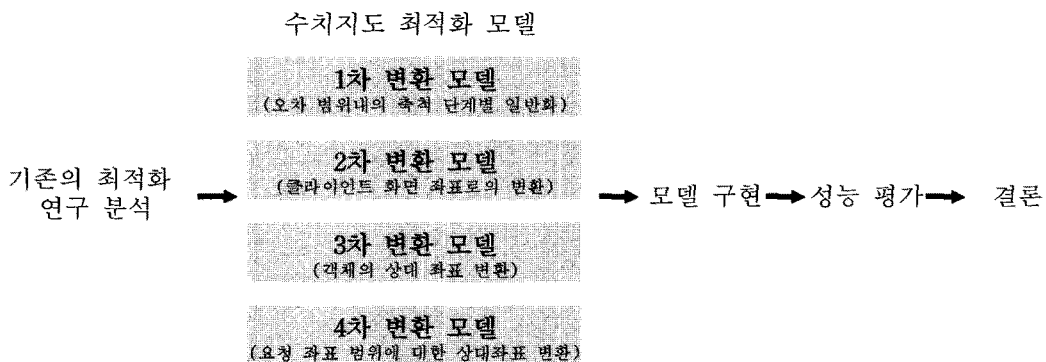
Keywords : mobile GIS, vector data, data-reduction

1. 서론

기존의 유선 환경에서는 정적인 서비스만을 해왔던 반면, 최근 들어 무선 모바일 기술의 발달로 이동성을 기반으로 하는 동적인 서비스의 요구 사항이 커지게 되고, 휴대 무선 모바일 기기의 대중화로 인해 정적인 서비스 외에 동적인 위치와 결합된 형태의 서비스가 가능해졌다. 하지만 이런 이동성을 기반으로 하는 모바일 GIS는 전송 데이터 크기, 클라이언트의 단말기 화면 해상도 및 크기, 응답 시간, 무선 단말기 자원, 무선 대역폭 등의 제약이 따른다. 한편 모바일 GIS 서비스에서 가장 중요한 사항은 사용자에게 보내어질 정보를 정확하게 전달하면서 정보의 양을

가능한 줄이는 기술이다. 이러한 문제를 해결하기 위해서 여러 연구가 활발히 진행되고 있으며 현재 일부 상용화 된 제품이 있으나 아직 표준화가 되지 않아 GIS 서비스를 개발하는 업체마다 개별적으로 이러한 제약을 극복하기 위한 기술을 연구하고 있다.

본 연구에서는 기존의 공간 데이터에 관한 경량화 연구에 대한 분석을 통하여 문제점을 도출하였다. 이러한 문제점을 극복하고 서비스에서 요구되는 정밀도를 고려한 네 단계의 수치 지도 경량화 모델을 정의하였고 이는 다음과 같다. 1차 변환 모델로는 오차 범위내의 축척 단계별 일반화, 2차 변환 모델로는 클라이언트 화면 좌표로 변환, 3차 변환 모델로는 객체의 상대 좌표 변환, 4차 변환 모델로는 요청



[그림 1] 연구의 흐름

좌표 범위에 대한 상대 좌표 변환이다. 정의된 모델을 시험하기 위하여 강남구 지역의 공간 데이터를 추출하였고 1차 변환 모델을 적용하기 위하여 ESRI사의 ArcInfo를 사용하여 데이터를 경량화 하였으며 2차 변환 모델, 3차 변환 모델, 4차 변환 모델로 공간 데이터를 경량화하기 위하여 JAVA 언어를 사용하여 구현하였다. 구현된 결과를 평가하기 위하여 기존에 유선 환경에서 서비스되고 있는 GIS 엔진에 본 논문의 모델을 적용하여 요청되는 공간 데이터의 크기, 초당 처리 개수, 하나의 요청에 대한 처리 시간을 측정하여 비교 분석하고 분석된 결과를 토대로 결론을 도출하였다. 본 연구의 흐름은 [그림 1]과 같다.

2. 관련 연구

2.1 유선 GIS 엔진

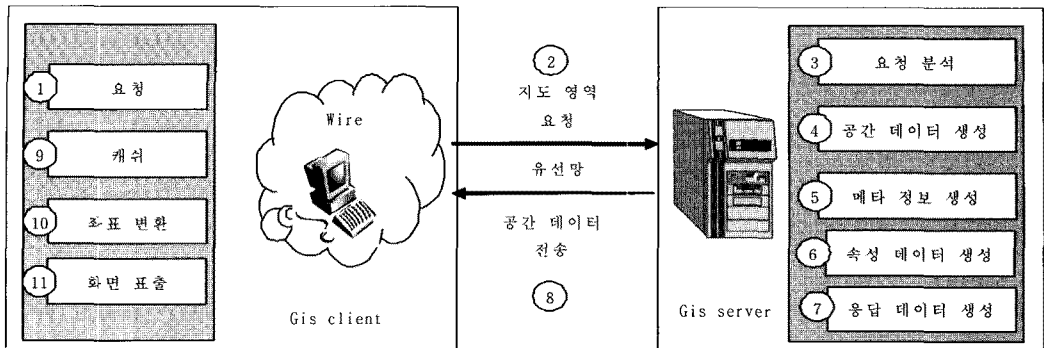
웹지도서비스(WMS:Web Map Service)는 인터넷 웹서비스와 GIS를 연동하여 지리 정보를 포함한 교통, 문화, 관광, 마케팅 정보 등 다양한 부가 정보를 제공하는 인터넷 서비스를 말한다. 이러한 서비스들은 여러 가지 형태의 콘텐츠들과 결합하고 있으며 최근 들어 LBS/텔레매틱스 서비스들과 연계된 활발한 연구가 진행되고 있다. 이러한 각각의 시스템과 원활한 결합을 위하여 국외에 대표적인 표준화 단체

인 OGC에서는 이러한 표준화 연구를 하고 있다.

OGC(Open Geospatial Consortium)는 웹 환경에서의 공간 데이터 서비스에 대한 표준을 발표하였고 벡터 데이터를 전송하기 위한 XML기반의 GML을 개발하였으며 이를 표출하기 위한 그래픽 기술로는 SVG¹⁾를 사용하였다. OGC 시스템에서는 다른 시스템간의 상호 운용이나 정보 공유를 위하여 데이터의 표준화를 추구하고 있으며 요청과 응답의 메시지 형식, 공간 정보 데이터들을 규정하고 있다.

대부분의 Web GIS 솔루션들은 클라이언트/서버로 구분된 형태의 구조로 보다 빠른 서비스를 구현하기 위하여 업체별로 특화된 공간 데이터 포맷을 사용하고 있으며 자체 솔루션을 개발하고 있다. 서버는 클라이언트의 요청 영역에 해당하는 공간데이터와 메타 데이터를 생성하여 응답하는 역할을 한다. 서버에서는 대용량 지리정보 데이터베이스를 관리하고 응답하기 위한 공간데이터 압축 기법 및 공간 인덱싱 기법을 사용하고 있으며, 공간 인덱싱 기법으로는 R-TREE에서 변형된 형태의 기법을 가장 많이 사용되고 있다. 클라이언트는 서버부터 응답받은 공간 데이터를 서버와 마찬가지로 공간 인덱싱 기법으로 저장을 하고 상대적으로 크기가 큰 데이터를 주고받기 때문에 효과적인 지도 제어를 위한 객체 재사용 기법이 사용되고 있다(김정준 2005, 윤재준 외 2002).

1) SVG(Scalable Vector Graphics)란 XML의 개방성, 상호 운용성 등의 장점을 벡터 그래픽에 모두 수용하였고, 다른 XML 언어들과 결합하여 다양한 응용의 개발이 가능하므로 광고, 전자상거래, 프로세스 컨트롤, 지리정보, 교육 등 그래픽이 많이 사용되는 분야에 적용된다.



[그림 2] 데이터 요청 흐름도

시스템 간의 데이터 요청 과정은 [그림 2]와 같다. 클라이언트가 통신망을(②) 통하여 서버로 데이터를 요청하면(①) 서버에서는 이를 분석하여(③) 요청 영역에 해당되는 데이터를 각각 공간 데이터 관리기(④), 속성 데이터 관리기(⑥) 메타 데이터 관리기(⑤)를 통하여 데이터를 추출한 후 응답 데이터를 생성하고(⑦), 이를 다시 통신망을(⑧) 통하여 응답 데이터를 클라이언트로 전송을 한다. 클라이언트는 전송받은 데이터를 캐쉬 관리기(⑨)를 통하여 공간 데이터, 속성 데이터, 메타 데이터를 저장하고 공간 데이터를 화면에 표출하기 위하여 좌표 변환(⑩)을 한다. 마지막으로 변환된 공간 데이터를 단말기의 출력기를 통하여 화면에 출력한다(⑪). 이 시스템을 제한된 모바일 환경에 적용하기 위해서는 무선망의 데이터 전송속도가 유선망과 비교하여 상대적으로 느리기 때문에 응답 데이터 생성에서 발생하는 데이터 크기를 최소화 하여야 한다. 또한 유선에서 사용되는 단말기의 메모리 크기에 비하여 무선 단말기의 메모리 크기가 상대적으로 작기 때문에 데이터 크기를 축소해야만 한다. 본 논문에서는 이러한 데이터를 최

소화하기 위한 기법을 제시하고 있다.

2.2 공간 데이터 경량화관련 기존 연구

일반적으로 GIS 엔진에서 사용되는 공간 데이터의 경량화는 크게 공간 데이터 일반화 기법과 공간 데이터 포맷 경량화 기법으로 나눌 수 있다. 공간 데이터 일반화 기법이란 대축적의 데이터가 소축적에서 사용자가 인지할 수 있는 범위 내에서 적합하게 표현될 수 있도록 데이터의 공간적 형태를 단순화시키는 기법이며, 공간 데이터 포맷 경량화 기법이란 서비스 환경에서 주고받는 데이터의 형태를 변형시키는 기법이다.

일반화의 유형은 연구자들에 따라 다양하게 나누지만, 보통 제거(elimination), 축약(collapse), 단순화와 완만화(simplification and enhancement), 선택(selection), 과장(exaggeration), 대표화(typification), 연결 관계(connectivity), 고유성 유지(uniqueness)의 8가지로 분류된다(김남신, 2003). 이러한 개념을 토대로 지도 축소·편집 자동화 시스템 개발(국토지리정보원, 2003)에서는 수치지도(Ver.2.0)를 기반으로 하는 자동지도제작 시스템을 구

축하였다. 이 시스템은 대축척의 1/5,000 수치지도를 기반으로 1/25,000 수치지도를 생성하기 위해 지도 일반화를 통하여 자동으로 변환하기 위한 시스템을 개발하는데 목적이 있다. 시스템은 크게 3단계로 구분되며 다음과 같다. 첫째, 원활한 축소·편집을 위하여 레이어를 삭제 및 분류하며, 둘째, 분류된 레이어를 단선처리, 축약, 정리, 집단화, 확대, 단순화, 완만화를 통하여 단순화 한다. 마지막으로 셋째, 일반화를 위하여 분류되어진 레이어를 합하여 1/25,000 수치지도를 생성하게 된다. 이러한 3단계를 통하여 지도 데이터 일반화를 수행하였고 1/5,000 수치지도에 비하여 약 최소 0.88%에서 최대 85% 정도로 데이터의 크기가 작아진다고 보고하고 있다.

김미란(2003)의 연구에서는 유선 GIS 엔진에서의 경량화를 위한 공간 데이터 포맷을 소개하고 있다. 서버와 클라이언트가 주고받는 데이터 좌표를 표현하기 위하여 기존 엔진에서는 x 좌표, y 좌표 총 16byte 나 8byte로 정의하였으나, 여기에서는 2byte 형태로 변환하여 데이터를 경량화하고, 일반화 기법과 포맷 경량화 기법을 클라이언트가 데이터를 요청할 때마다 응답 결과를 생성할 때 각각 변환기를 두어 변환하는 것으로 시스템을 정의하였다. 그러나 클라이언트가 데이터를 요청할 때마다 많은 양의 공간 데이터를 일반화와 필터링을 수행할 경우 서버에서 발생하는 오버헤드가 상당 부분 커질 것이고, 요청하는 영역의 축척마다 이러한 경량화 과정으로 변환된다면 일관성 없는 데이터로 인하여 클라이언트에서 공간 데이터 캐시를 하기 힘든 구조로 될 것이다. 또한 하나의 좌표

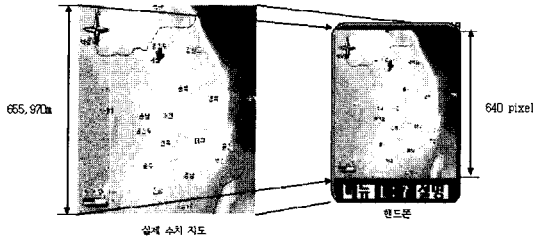
를 2byte 형태로 보내는 것도 1byte가 표현할 수 있는 수치의 범위가 0~255이고 클라이언트의 단말기의 화면 해상도가 최근 들어 최고 가로×세로 640×480pixel 이상의 기기도 출시되고 있어 화면의 해상도가 255 pixel을 넘는 단말기의 화면에 좌표를 표현하는데 문제가 발생한다. 따라서 본 연구에서는 이러한 서버에서 발생하는 오버헤드와 클라이언트 객체 재사용 기법, 그리고 단말기의 화면 해상도에 대한 문제점을 극복하기 위하여 좀 더 효과적인 일반화 기법 및 공간 데이터 포맷 경량화 기법에 관하여 제시하였다.

3. 모바일 환경의 수치지도 경량화 기법

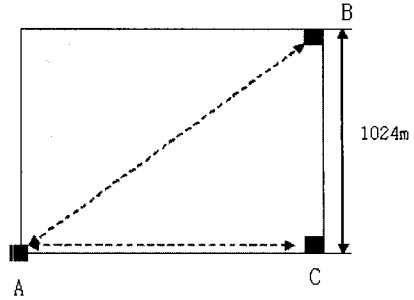
3.1 오차 범위 내의 축척 단계별 일반화

클라이언트의 공간 데이터 요청마다 결과 데이터에 대한 일반화를 수행할 경우 서버에 발생하는 오버헤드와 클라이언트의 공간 데이터 캐시가 문제가 된다. 이를 극복하기 위하여 서비스 이전에 축척 단계별로 구분하여 각 축척 단계에 대한 일반화 방법을 제시하려고 한다. 여기서의 일반화란 기존의 종이 지도를 구축하기 위한 일반화의 개념에 모바일 단말기의 해상도에서 발생할 수 있는 오차 범위에 대한 일반화 개념이 포함되어 있다.

전국 지도를 표현하는 축척 단계를 모바일 단말기에 표출했을 경우 [그림 3]과 같이 표현될 수 있고 그림 왼쪽의 세로 거리가 655,970m인 실제 지도 영역을 모바일 단말기의 세로 해상도 640pixel에 표현할



[그림 3] 단말기 화면의 오차 범위



[그림 4] 1 pixel의 거리 오차

경우 1pixel의 가로 폭의 실제 거리는 1024m가 된다. [그림 4]와 같이 가로×세로 1pixel의 크기에서 실제 좌표 A, B, C가 그림과 같이 위치할 경우 이들은 화면상에서 모두 같은 좌표 값을 갖는다. 하나의 픽셀에서 발생할 수 있는 최대 오차 범위는 픽셀의 좌측 하단 모서리에서 우측 상단 모서리와의 거리가 된다. 그러므로 전국 지도를 표현하는 축척 단계에서는, 거리가 1449m 이하의 지형 요소는 하나의 점으로 밖에 표현할 수 없다는 것을 알 수 있다.

$$\begin{aligned}
 & \text{좌표 } A, B \text{에 대한 거리} \\
 & = \sqrt{(y\text{축 픽셀의 실제거리})^2 + (x\text{축 픽셀의 실제거리})^2} \\
 & = \sqrt{1024^2 + 1024^2} \\
 & = 1449m
 \end{aligned}$$

이러한 오차 범위를 적용하여 서비스가 요구되는 정밀도에 따라 여러 개의 축척 단계를 구분할 수가 있다. 축척 단계의 구분 및 결정에 앞서 우선 서비스의 공간적 범위와 정밀도의 분석이 선행되어야 한다. 공간적 범위와 정밀도에 따라 단말기에서 지도를 표출할 최대 축척과 최소 축척을 결정하고 결정된 최대/최소 축척 사이의 중간 축척 단계는 몇 개로 구분할 것인가를 결정해야만 한다. 중간 축척 단계는 일

정한 비율로 나누어서 구분하고 좀 더 정밀도를 요구하는 축척 영역은 좀 더 많은 축척 단계를 두어 구분할 수 있다.

3.2 클라이언트 화면 좌표로의 변환

클라이언트 화면 좌표로의 변환은 객체의 상대 좌표 변환의 효율을 높이기 위한 전 단계로 서버에 저장되는 데이터의 수치 값을 각각의 축척 단계에서 표현할 수 있는 정밀도의 수치 값으로 변환하는 것을 말한다. 이것은 각각의 축척 단계에서 클라이언트 단말기의 해상도로 그 이상의 정밀도를 표현할 수 없기 때문에 해당 축척에 적합한 정밀도로 좌표값을 표현하는 과정이다. <표 1>은 축척 단계의 공간 객

<표 1> 축척 단계 별 클라이언트 화면 좌표 범위

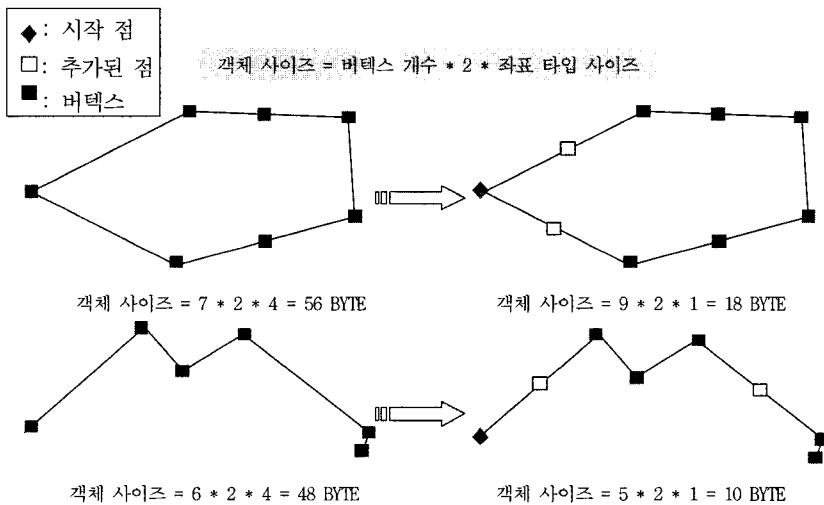
축척 단계	클라이언트 화면좌표 최대값	실제 거리(m)
1	640	655,970
2	1920	655,970
3	5760	655,970
4	17280	655,970
5	34560	655,970
6	103680	655,970
7	311040	655,970
8	933120	655,970

체들이 변환될 좌표의 범위를 나타낸다. 실제 거리는 우리나라 전국을 포함하는 가로×세로 길이 중 큰 값을 나타내며 클라이언트 화면 좌표는 공간 객체 수치의 최대값을 의미한다. 모바일 단말기의 최대화면 해상도를 가로×세로 640×640pixel로 가정했을 경우 축척 1단계의 공간 객체의 좌표 범위는 0~640 내의 수치 값으로 변환되며 축척 8단계의 공간 객체의 좌표 범위는 0~933120 내의 수치 값으로 변환된다. 만약 모바일 단말기의 화면 해상도가 앞절에서 클라이언트 화면 좌표 변환을 위해 가정된 최대 해상도(예, 640×640pixel)보다 낮은 해상도를 갖고 있다면 화면에 표출할 수 있는 정밀도가 이보다 낮기 때문에 객체에 대한 왜곡 현상은 없을 것이다. 즉 단말기의 해상도와 좌표 변환에 가정된 최대 해상도(예, 640×640)의 비율을 이용하여 왜곡 없이 화면에 표출을 할 수 있다.

3.3 객체의 상대 좌표 변환

현재 네이버에서 서비스되고 있는 GIS 엔진의 통신 프로토콜 포맷에서 공간 데이터의 수치 영역은 PointX, PointY로 Point Count가 커질수록 공간 객체의 크기가 커진다. PointX, PointY의 데이터 타입을 -128~127의 수치 범위를 표현할 수 있는 BYTE 형태로 변환한다면 공간 객체의 크기를 상당히 줄일 수 있게 된다. 시작 포인트를 제외한 연결 포인트의 좌표 값을 이전 포인트에 대한 상대 값으로 변환한다. 이때 상대 좌표 값이 1byte로 표현할 수 있는 값의 범위를 넘게 되면 중간에 포인트를 추가하여 모든 연결 포인트 값을 1byte 형태로 변환할 수 있다. [그림 5]에서 폴리곤과 폴리라인 객체를 왼쪽에 절대 좌표에서 오른쪽의 상대 좌표로 변환할 경우 공간 데이터의 사이즈가 작아지는 것을 확인할 수 있다.

객체의 상대 좌표로 변환하기 이전에



[그림 5] 공간 객체의 상대 좌표 변환

반드시 위에서 논의되었던 클라이언트 화면 좌표의 변환과정을 거쳐야만 한다. 예를 들어 축척 1단계에서 변환되지 않은 좌표 값을 상대 좌표로 변환을 한다면 중간에 1byte로 표현할 수 없어서 추가가 되는 포인트가 개수가 상당히 증가할 것이고 오히려 데이터의 크기가 증가하는 결과가 발생하게 된다. 이것으로 소축척 단계의 수치 좌표의 범위가 대축척 단계보다 작기 때문에 추가되는 포인트 수가 많아지고 변환에 대한 효율이 작아지는 것을 알 수 있다.

4. 시스템 구현 및 성능 평가

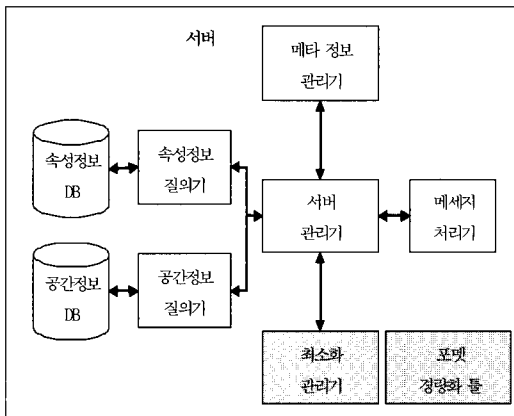
4.1 시스템 구현

모바일 환경의 GIS 엔진의 시스템 구조는 지도 데이터베이스로부터 벡터 데이터와 속성 데이터를 선택하여 클라이언트로 전송하는 서버 모듈과, 서버로부터 전송받

은 벡터 및 속성 데이터를 단말기 화면에 출력하는 클라이언트 모듈로 나뉜다. [그림 6]과 [그림 7]은 각각 서버와 클라이언트의 구성을 보여주고 있다.

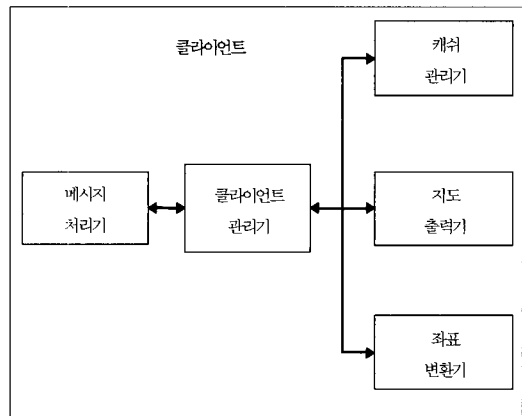
서비스의 공간적 범위는 강남구 행정구역을 선택하였고 단말기의 최대화면 해상도를 가로×세로 640×640pixel로 정하였으며 축척의 단계는 총 8단계로 구분하였다. 구분된 단계 중에 축척 8단계에서 축척 4단계까지의 공간데이터를 경량화하였다. 최소 축척 단계에서의 1pixel의 오차 범위는 1449.50m이고 좌표의 최대 수치 값은 640이며 최대 축척 단계에서의 1pixel의 오차 범위는 0.99m이고 좌표의 최대 수치 값은 933120이 된다.

공간데이터 경량화 과정은 [그림 8]과 같이 4단계의 순서로 진행을 하였다. 오차 범위 내의 축척 단계별 일반화 과정은 ESRI사의 ArcInfo를 사용하였고 클라이언트 화면 좌표 변환, 객체의 상대 좌표 변환 과정은 [그림 6]에서 설명한 바와 같이



※ 개발 툴 : JAVA 2 SE(1.4.2)
운영체제 : Red Hat Linux 7.0

[그림 6] 서버 구성도 및 환경

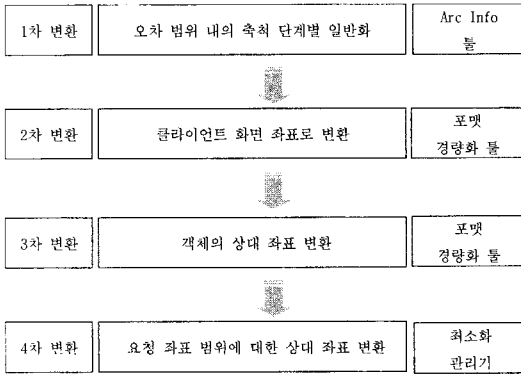


※ 개발 툴 : WINDOWS CE 4.0
운영체제 : Pocket 2003 Emulator

[그림 7] 클라이언트 구성도 및 환경

포맷 경량화 틀로 변환하였다. 마지막 요청 좌표 범위에 대한 상대 좌표 변환은 최

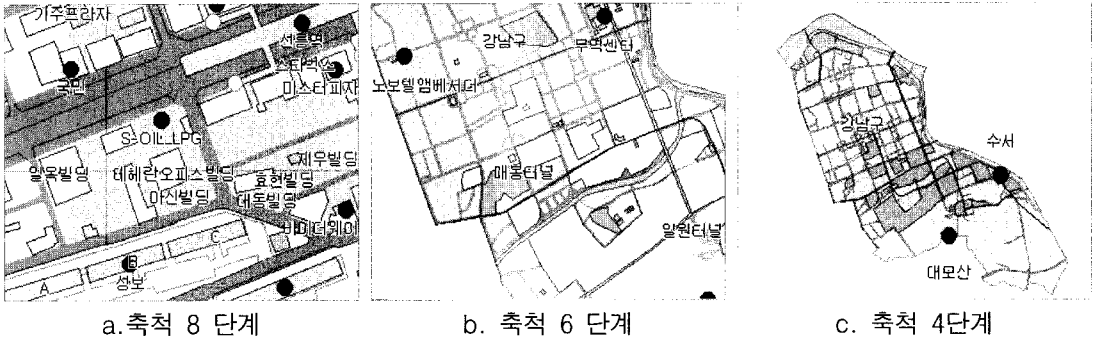
소화 관리기로 변환하였다. [그림 9]는 클라이언트에서 축척 단계별로 호출되는 지도 화면이다.



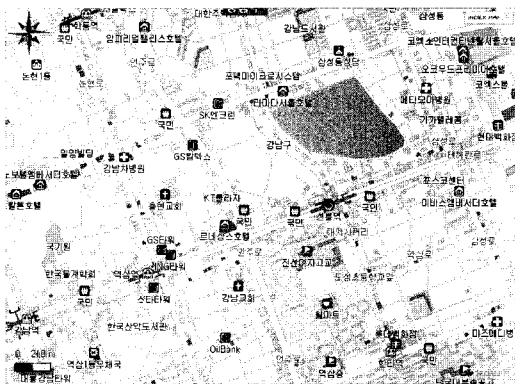
[그림 8] 벡터 데이터 변환 과정

4.2 성능 평가

성능 평가는 현재 네이버에서 서비스되고 있는 GIS 엔진과 본 연구에서 제시한 공간 데이터의 경량화 기법을 적용한 엔진을 비교하여 테스트하였다. 성능 테스트를 위하여 이전에 두 개의 시스템에서 동일한 영역을 선택하였는데, [그림 10]은 유선에서 현재 서비스 되고 있는 지도 화면



[그림 9] 축척 단계별 지도 화면



[그림 10] 유선에서 서비스 되고 있는 GIS 엔진의 지도 화면(출처:naver)



[그림 11] 경량화가 적용된 GIS 엔진의 지도 화면

이고 [그림 11]은 경량화가 적용된 지도 화면이다. 테스트는 첫째, 동일한 하드웨어 사양의 시스템에 GIS 엔진을 설치하고 선택된 영역을 각각 한번 씩 요청하여 서버에서 응답하는 데이터의 크기를 확인하고, 둘째, [그림 12]와 같이 처음 2분 동안 5개의 클라이언트가 계속해서 요청하고 2분이 지나면 5개의 클라이언트를 추가로 요청하는 방식으로 진행하였다. 2분마다 테스트 장비의 CPU 사용량, 서버의 총 응답 개수, 각각 요청에 대한 평균 응답 시간을 측정하였으며 테스트 장비의 CPU 사용량이 100%에 달했을 때 요청을 중단하였다.

[그림 10]과 [그림 11]의 지도 영역을 요청하여 서버에서 응답한 데이터의 총 크기를 측정한 결과는 <표 2>와 같고 여기에서 경량화를 적용한 시스템의 데이터 크기가 기존의 시스템 보다 57% 정도 줄어들었음을 확인하였다.

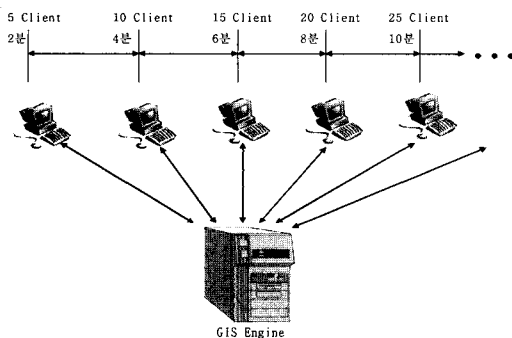
[그림 13]은 요청하는 클라이언트 증가에 따른 CPU의 사용량을 도식화한 것이다. 초기 5개의 클라이언트가 요청한 경우 기존 GIS 엔진의 CPU 사용량은 평균 20%,

경량화가 적용된 GIS 엔진의 CPU 사용량은 평균 10%임이 확인되었고 요청이 증가함에 따라 전자의 CPU 사용량이 상대적으로 후자보다 CPU 자원을 많이 사용하는 것을 확인하였다. 또한 요청하는 클라이언트의 개수가 30개에 도달했을 때 전자의 CPU의 사용량은 100%에 도달한 반면 후자의 CPU 사용량은 60% 정도였으며 요청 클라이언트가 66개가 되었을 때 비로소 CPU 사용량이 100%에 도달 되는 것이 측정되었다.

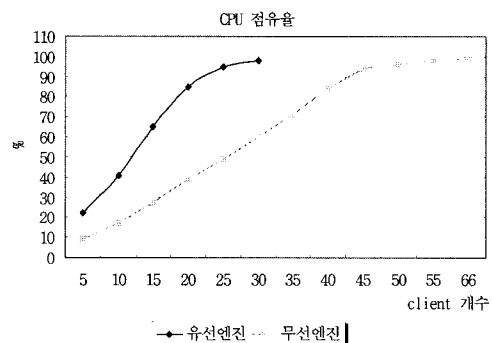
초당 응답하는 개수의 평가 결과는 [그림 14]와 같다. 초기 5개의 클라이언트가 요청되었을 경우 기존 GIS 엔진은 11개, 경량화가 적용된 GIS 엔진은 13개로 둘 다 비슷한 수치가 나왔고 요청이 증가함에 따라 서버에서 응답하는 개수는 늘어났으며 전자는 CPU 사용량이 100%되는 시점인 요청 클라이언트 개수 30개가 되었을

<표 2> 동일한 영역 요청시 데이터 크기

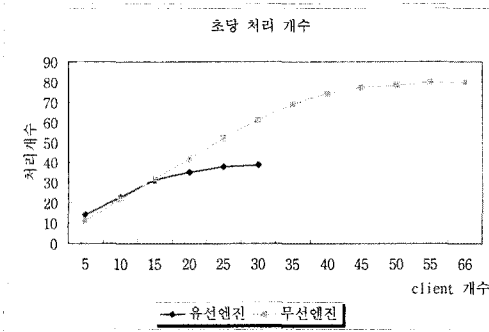
	Sundo GIS엔진	최소화 적용엔진	축소 비율(%)
크기 (byte)	168035	71364	57



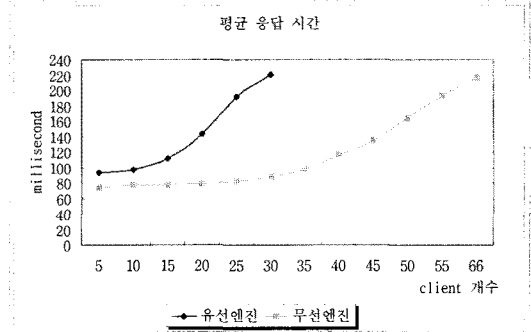
[그림 12] 성능 평가 과정 모식도



[그림 13] CPU 점유율



[그림 14] 초당 처리 개수



[그림 15] 평균 응답 시간

때 초당 처리 개수가 30개였고 더 이상 늘어나지 않는 것이 확인되었다. 이에 비해 후자는 요청하는 클라이언트 개수가 55개가 되었을 시점에 초당 처리 개수가 80개였고 비로소 증가량이 없음이 측정되었다. 이는 경량화로 인하여 상대적으로 데이터의 크기가 줄어들어 CPU의 자원을 적게 사용함으로써 좀 더 많은 요청 건수를 처리할 수 있음을 확인할 수 있다.

요청 하나의 평균 처리 시간 결과는 [그림 15]와 같다. 초기 5개의 클라이언트가 요청 되었을 경우 기존 GIS 엔진은 0.10초, 경량화가 적용된 GIS 엔진은 0.08초이며 후자가 상대적으로 빠르게 응답했다. 요청이 증가함에 따라 평균 처리 시간이 전자는 빠르게 증가하여 요청 클라이언트의 개수가 30개일 경우 평균 처리 시간이 0.22초에 도달하는 반면, 후자는 요청 클라이언트 개수가 30개 되었을 때 비로소 증가하기 시작했으며, 요청 클라이언트 개수가 66개일 때 평균 처리 시간이 0.22초에 도달하는 것을 확인하였다.

성능 평가 결과 공간 데이터의 경량화로 기존의 GIS 엔진에서 응답하는 데이터의 크기가 절반 이상 줄었고, 이로 인하여

서버의 CPU 사용량이 상대적으로 줄어들었으며 줄어든 CPU 자원으로 인하여 초당 처리하는 개수가 늘어나는 것이 확인되었다. 또한 요청에 대한 평균 처리 속도도 기존 엔진에서 보다 향상되는 것이 확인되었다.

5. 결 론

본 연구에서는 모바일 환경에서 전자지도 서비스를 하는데 있어서 여러 제약 사항들을 극복하기 위한 경량화 기법을 제시하였다. 서비스에서 요구되는 정밀도를 분석하고 결정된 정밀도의 범위 내에서 공간 데이터를 경량화하는데 목적이 있으며, 그 과정은 다음과 같이 요약된다. 첫째, 축척 단계를 나누어 각 단계별 오차 범위 안에서 일반화 과정을 거쳐 공간 데이터의 크기를 줄인다. 둘째, 공간 데이터의 수치 좌표를 화면 좌표 값으로 변환하여 공간 객체의 수치 좌표값을 기존보다 작은 좌표값으로 변환한다. 셋째, 공간 데이터의 시작 포인트를 제외한 나머지 연결 포인트들을 2byte(x,y 각각 1byte)로 표

현할 수 있는 수치 범위 내에서 이전 좌표의 상대 좌표로 변환함으로써 데이터의 크기를 줄인다. 넷째, 클라이언트가 요청하는 좌표 영역을 255×255 크기의 격자 형태로 나누어서 각각의 격자 안에 포함되는 공간객체의 좌표 값을 2byte 형태로 변환한다.

본 연구에서 제시한 경량화 기법을 검증하기 위하여 기존에 웹 환경에서 서비스되고 있는 GIS 엔진과 본 논문의 경량화 알고리즘을 적용시킨 엔진과의 비교 성능 테스트를 통하여 정량적인 결과 값을 도출하였다. 테스트 결과 데이터 크기는 57% 감소하여, 기존 GIS 엔진과 비교하여 본 연구의 경량화 기법이 약 두 배 정도 향상이 있었음을 확인하였다. 성능 평가 결과 본 연구의 방법론은 재의 모바일 GIS 서비스 환경에서 서버 부하 감소, 응답 데이터의 경량화로 인한 데이터압축 과정 생략, 기존 엔진대비 빠른 전송 속도 및 화면 표출, 일관된 객체 재사용 기법 적용 등의 의미를 지닌다.

모바일 GIS는 민간 및 공공 부분에서 ITS, LBS, 텔레매틱스 기술과 함께 이들을 지원하기 위한 기반 기술로서, 그 수요와 활용 분야도 점차 다양화 되고 있는 핵심 기술 부분이다. 이러한 모바일 GIS 서비스를 위한 관련 기술의 개발이 요구되고 있어 본 논문에서 제시한 모바일 환경의 제약사항 극복을 위한 벡터 데이터 경량화 기법은 실질적으로 적용될 수 있는 연구라 할 수 있다.

감사의 글

본 연구는 건설교통부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C04)에 의해 수행되었습니다.

참고문헌

- 김미란, 2003, “모바일 GIS를 위한 클라이언트/서버 시스템 연구”, 부산외국어대학교 박사학위논문
- 김정준, 2005, “공간 모바일 장치를 위한 공간 MMDBMS의 설계 및 구현”, 건국대학교 석사학위논문
- 배상근, 박영무, 김병국, 모바일 맵핑시스템에서의 실시간 지리정보 전송을 위한 연구, 한국GIS학회지 제13권 제1호, 2005. 4, pp. 91~101
- 윤재준, 이근호, 한기준, 2002, “Shareware DBMS를 이용한 모바일 GIS의 개발”, 개방형GIS학회, 제2권 제1호 pp.25~37
- 국토지리정보원, 2003, “지도 축소·편집 자동화 시스템 개발”
- 네이버 지도 서비스, <http://www.naver.com>
- Woo Je Chun, Yong Jin Joo, Kyung Ky Moon, Yong Ik Lee, Soo Hong Park, A GIS Vector Data Compression Method Considering Dynamic Updates, The Journal of GIS Association of Korea, Vol. 13, No. 4, pp. 355-364, December 2005

접수일 (2008년 6월 16일)

수정일 (2008년 7월 17일)

게재확정일 (2008년 7월 21일)