

모바일 디바이스에서 GML 가시화를 위한 분할 및 캐싱 기법

송은하[†], 박용진^{**}, 한원희^{***}, 정영식^{****}

요 약

본 논문은 모바일 디바이스에 GML을 효율적으로 공급하고 가시화하기 위한 맵 분할 및 캐싱 기법을 제안하고, 이를 적용한 GridGML을 구현한다. GridGML은 GML의 가장 큰 단점인 가중성을 극복하기 위해 GML 속성들 중 맵 가시화 핵심 부분만을 추출 및 클래스 객체로 경량화 한다. GridGML은 모바일 디바이스에 실시간으로 맵을 공급하고 가시화하기 위해 모바일 디바이스 가시영역을 기준으로 분할 관리하며, 분할 영역은 전송상의 이점을 위해 바이트화 하여 전송한다. 그리고 모바일 디바이스에서는 수신된 분할영역을 조합한 후 모바일 디바이스에서 표현하는 영역에 따라 4개의 가시영역으로 재분할 및 가시화를 한다. 또한, 자원의 효율적 운영을 위해 이전에 전송받은 맵의 중복성을 고려한 캐싱 알고리즘을 적용하여 관리한다. 맵의 객체 밀집 지역에 대해서는 전송시간의 지연을 방지하기 위해 적응적 맵 분할 메커니즘을 제안하여 전송시간을 일정함을 유지한다.

Partition and Caching Mechanism for GML Visualization on Mobile Device

Eun-Ha Song[†], Yong-Jin Park^{**}, Won-Hee Han^{***}, Young-Sik Jeong^{****}

ABSTRACT

In this paper, we developed GridGML for efficiently supplying a GML and visualizing the map with partitioning map and caching method to a mobile device. In order to overcome the weighting of a file, which is the biggest weakness of a GML, GridGML extracts only the most necessary parts for the visualization of the map among GML attributes, and makes the file light as a class instance by applying an offset value. GridGML manages a partition based on the visualization area of a mobile device to visualize the map to a mobile device in real time, and transmits the partition area by serializing it for the benefit of transmission. Also, the received partition area is compounded in a mobile device and is visualized by being partitioned again as four visible areas based on the display of a mobile device. Then, the area is managed by applying a caching algorithm in consideration of repetitiveness for a received map for the efficient operation of resources. Also, in order to prevent the delay in transmission time as regards the instance density area of the map, an adaptive map partition mechanism is proposed for maintaining the transmission time uniformly.

Key words: GIS(지리정보시스템), GML(지리 마크업 언어), map visualization(맵 가시화), caching algorithm(캐싱 알고리즘), adaptive map partition mechanism (적응적 맵 분할 기법)

※ 교신저자(Corresponding Author):정영식, 주소: 전북 익산시 신용동 344-2(570-749), 전화: 063)850-6746, FAX: 063)856-8009, E-mail: ysjeong@wku.ac.kr

접수일: 2008년 1월 10일, 완료일: 2008년 5월 14일

[†] 준회원, 원광대학교 전기전자 및 정보공학부 전임강사
(E-mail: ehsong@wku.ac.kr)

^{**} 준회원, 원광대학교 컴퓨터공학과 석사

(E-mail: yjpark1@wku.ac.kr)

^{***} 원광대학교 컴퓨터공학과 석사과정

(E-mail: samsung@wku.ac.kr)

^{****} 정회원, 원광대학교 전기전자 및 정보공학부 교수

※ 본 논문은 2006년 산학협동재단에서 지원하는 학술연구비 지원으로 수행되었음

1. 서론

최근 모바일 기술의 급성장은 우리의 생활에 편리함을 가져다주고 있다. 컴퓨터 앞에 앉아서만 가능하던 일들이 포터블한 모바일 장치로 인해 빠르게 변화하는 정보에 대해 시시각각 대응할 수 있게 되었다. 좀 더 효율화된 정보처리가 가능하게 됨에 따라 이와 관련된 다양한 지원 모델과 관련 응용 프로그램들이 개발되고 있다. 이는 비단 특정분야에 한정하지 않고 갈수록 세분화, 전문화되어가는 지리정보서비스에서도 이 모바일 기술은 현시대를 대표하는 하나의 아이콘으로 부각되고 있다. 하지만 현재 LBS는 기반이라 할 수 있는 맵 데이터에 대해 각 개발 업체마다 각기 다른 포맷을 사용하여 서비스하고 있어 맵 데이터들 간에 자유로운 활용과 통합 및 응용이 불가하다. 따라서 공통교환 포맷 표준이 필요하게 되었는데 OGC(Open GIS Consortium)에서는 공간적 또는 비공간적 속성을 지닌 지리사상(Feature)을 포함하는 지리정보의 전달과 저장을 위한 지리정보 인코딩을 뜻하는 XML 기반의 GML(Geography Markup Language) 명세를 제시하였다[1-4]. GML은 확장성과 유연성이 뛰어나 좀 더 효율적인 데이터 처리 및 관리가 가능하다. 그러나 표준명세인 GML은 ASCII 포맷으로써 기존 바이너리 형식의 포맷에 비해 파일이 무겁다는 단점이 존재한다. 이는 데이터 처리 및 관리에 있어서 시스템의 고사양이 필수가 되고 특히 모바일로의 서비스는 네트워크상에 존재하는 트래픽에 의해 맵 전송이 용이하지 못하며 전송된 맵에 대해서도 처리를 위한 모바일 디바이스의 성능적 한계를 극복하기 어렵다. 또한 기존 LBS는 맵 데이터를 모바일 기기에 탑재하는 방식을 사용하는데 이는 도로 및 건물들이 시시각각 변화하는 최근 상황에 비추어 볼 때 잦은 업그레이드를 요구하므로 사용자로부터 불편함을 야기한다.

이에 본 논문은 모바일 디바이스에 GML을 효율적으로 공급하고 가시화하기 위해 맵 분할 및 캐싱 기법을 제안하고, 이를 적용한 GridGML을 구현한다. GridGML은 서버의 맵 요청에 빠른 응답을 지원하기 위해 GML을 모바일 디바이스의 디스플레이 크기를 기준으로 분할하고 관리한다. 맵 요청시 모바일 디바이스의 위치가 포함된 영역이 선택되며 가시화에 필요한 부분만을 추출하며 데이터 압축을 하는

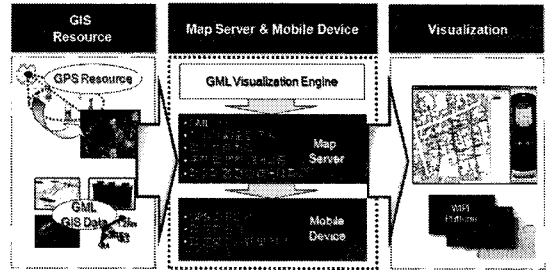


그림 1. 전체 시스템 개념도

필터링 과정의 수행으로 모바일 환경의 성능적 제약을 극복한다. 필터링된 분할 영역은 전송상의 이점을 위해 바이트화하여 전송한다. 모바일 디바이스에서는 WIPI 표준 모바일 플랫폼을 기반으로 맵의 실시간 가시화를 위해 서버로부터 수신된 분할영역을 조합한 후 디스플레이창의 표현 영역에 따라 네 가지 영역으로 분할 가시화를 하고, 자원의 효율적 운영을 위해 전에 전송받은 맵에 대해 캐싱 알고리즘을 적용하여 관리한다. 그림 1은 전체 시스템 개념도이다.

본 논문의 구성은 2장에서 관련연구에 대해 설명하고, 3장에서 본 논문에서 제안하는 동적 맵 분할 및 캐싱 기법에 대해 설명한다. 4장에서는 동적 맵 분할 및 캐싱 기법에 의해 설계하고 구현한 GridGML 설계 및 구현에 대해 기술하며, 5장에서 성능평가를 통해 본 논문에서 제안한 기법을 평가하고, 끝으로 6장에서 결론 및 향후 연구 과제를 기술한다.

2. 관련연구

2.1 GML

최근 GIS데이터들 중 현재 많은 국내 및 국외 이동 단말기용 맵 데이터로 사용하고 있는 GML은 OpenGIS 컨소시엄에 의해 정의된 지리정보의 저장과 호환을 위한 하나의 규약으로 도형 및 속성 정보를 갖는 Feature들을 포함하는 지리정보의 전송과 저장을 위한 XML 인코딩 기술이다[5,6]. GML은 지리정보를 XML로 인코딩 하는 과정에서 XML DTD에 기반하여 3가지 형태의 프로파일로 표현이 되는데 GML 1.0에서는 DTD의 한계성으로 인하여 GML 2.0부터 XML스키마를 기반으로 상호 확장이 용이하도록 3가지 스키마가 제공되었고, 현 GML 3.0에서는 공간 스키마의 능력이 향상되고 응용 스키마가 추가

되었다. 현재 GML은 우리나라 지리표준이다.

그러나 GML은 그 형식이 XML 기반의 ASCII 포맷으로 되어 있어 파일이 무겁다는 단점이 있다[7,8]. 이는 끝없이 그 성능이 발전해 나아가는 데스크톱 환경에서의 데이터 처리는 크게 문제되지 않으나 갈수록 모바일화 되어가는 현 시점에서 ASCII 포맷은 모바일의 환경상의 성능적 제약에 의해 크나큰 걸림돌이 된다. 본 논문에서는 GML의 가장 큰 단점인 파일의 가중성을 극복하고자 데이터들을 분할하여 관리할 뿐 아니라, 모바일에서의 데이터 처리를 위해 필터링 및 데이터 압축을 하고 전송 데이터에 대한 바이너리로 변환하여 전송한다.

2.2 LBS 기술 및 서비스

국내 위치기반서비스는 이동통신 3사인 SKT, KTF, LGT를 중심으로 개별적인 플랫폼을 보유하고 각사의 공급 전략에 따라 콘텐츠 및 서비스 제공자들이 통신망을 통한 서비스를 제공하고 있다. 국내 업체의 주요 서비스로는 Cell ID 기반의 “친구찾기”, “위치확인”, “주변정보” 서비스 등이 제공되고 있으며, “친구 찾기” 서비스의 경우 이동통신 3사간 연동 서비스가 되는 유일한 서비스이다. 2002년부터 GPS 기반의 서비스가 시작되어, KTF가 GPS 수신기를 장착한 어린이, 노약자용 안전관리 서비스(nGeleye)를 선보였다. SKT는 2002년 8월부터 GPS 폰을 출시, 기존의 NATE 포털과 연결하여 NATE GPS, NATE Drive 서비스를 제공하고 있다[9-12].

국내 위치기반 서비스는 이동통신 3사가 독자적인 플랫폼을 이용하여 개발 및 서비스를 실시하고 있어, 이동통신사업자, 콘텐츠 제공자 및 단말기 제조업자들 간의 상호 호환성 및 재사용이 어렵다. 이는 콘텐츠 및 단말기를 떠나서 가장 기반이 될 수 있는 맵 데이터의 표준을 따르지 않는 이동통신사들 간의 중복 투자 및 개발에 의해, 경제적 손실과 통신 시장의 국제 기술개발 우위의 선점에 한계가 있다. 또한, 이동통신 3사가 똑같이 적용하는 지도데이터 탑재방식은 수시로 변화하고 있는 도로상황이나 신규개통구간에 대해 동적으로 대응하기 힘들다. 더욱이 컴퓨터 활용능력이 떨어지는 50대 이후의 사람들로써는 업데이트의 한계성을 드러내기도 한다.

이에 본 논문에서는 맵 데이터의 표준인 OGC의 GML을 수용하고 맵 서버로부터 모바일 디바이스에

GML을 제공함으로써 위치기반서비스의 표준화를 실현하며, 지도 제공에 있어 실시간 전송 및 가시화를 위한 해결책을 제시한다.

3. 동적 맵 분할 및 캐싱 기법

본 논문은 디스플레이 크기가 120*160의 Aroma WIPI Emulator 환경을 기반으로 맵 서버와 모바일 디바이스간 맵 전송 및 가시화를 위해 분할 및 캐싱 기법을 설계한다. 디스플레이 화면에서 맵 표현영역은 줌 레벨 1일 경우 가로 세로 각각 60m를 표현 영역으로 한다. 또한 이동 속도의 경우 성인 남자가 4km/h의 속도로 걷는다는 가정하에 동적 맵 분할 및 캐싱 기법을 제안한다.

3.1 맵 분할 영역 추출

본 논문은 모바일 디바이스에 가시화를 목적으로 하고 있기 때문에 모바일 디바이스의 디스플레이 크기를 기준으로 4개의 영역으로 분할한다. 4개의 영역은 디스플레이 영역, 스크린 영역, 액티브 영역, 캐쉬 영역으로 구성된다. 디스플레이 영역은 모바일 디바이스의 화면 크기에 해당하는 영역이며, 스크린 영역은 맵을 그리고 있는 영역이고, 맵 재전송의 기준이 되는 영역, 액티브 영역은 맵 재전송에 따르는 지연 시간을 위한 예비 영역, 캐쉬 영역은 과거에 받아놓은 맵의 재활용을 위해 일정기간 저장해 놓는 영역이다. 그림 2는 가시 영역을 분할한 화면이다.

3.1.1 스크린 영역과 액티브 영역 추출

스크린 영역과 액티브 영역은 모바일 디바이스의

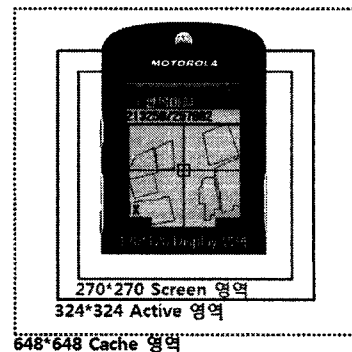


그림 2. 가시 영역 분할

디스플레이 영역을 기준으로 추출한다. 디스플레이 영역은 120*160 크기이나 인터페이스 부분을 빼 120*120 크기를 디스플레이 영역이라 정의한다. 디스플레이 영역은 줌 레벨이 1일 경우 표현 가능한 맵 크기는 가로, 세로 각각 60m이다. 여기에 성인 남성의 걷는 속도가 4km/h라고 가정하고, 맵의 재전송 타이밍을 30초로 잡고 30초간 재전송 없이 이동할 수 있는 영역 270*270(실제 맵 크기 : 135*135)을 스크린으로 추출한다. 스크린 영역을 벗어나 재전송이 이루어지면, 맵을 다운로드 하는 동안에 실시간 가시화를 위해 액티브 영역을 둔다. 액티브 영역은 스크린 영역 대비 120%의 영역으로 두고 있으며, 재전송에 대한 15초 정도의 지연시간을 보완할 수 있다. 서버로부터 맵을 전송받는 크기는 액티브 영역에 해당하는 크기를 전송받는다. 그림 3은 스크린 영역과 액티브 영역을 추출한 화면이다.

3.1.2 서버의 맵 분할

모바일 디바이스의 액티브 영역이 결정되면 서버의 맵 분할 기준이 결정된다. 이는 본 논문의 기준이 모바일 디바이스이므로 분할 방법이 결정된 후에 이에 맞는 최적의 크기를 맵 서버에서 전송해야 한다. 맵 분할은 모바일 디바이스로의 전송 시 Best Case와 Worst Case를 고려하여 분할한다. 맵 서버는 액티브 영역 대비 1/4의 경계값을 기준으로 분리한다. Best Case에는 4개의 분할 영역이 전송되고, Worst Case에는 9개 영역이 전송되도록 한다. 좀 더 세분화하여 맵을 분할할 경우 Best Case와 Worst Case의 차를 더 줄일 수는 있으나, 그럴 경우 맵의 줌 레벨 1에 포함되는 객체들의 수가 너무 적어지는 경우가 생길 수 있고 큰 도로의 경우에는 객체가 포함되지 않는 경우가 생길 수 있다. 이에 액티브 영역 대비 1/4의 경계값이 객체들이 고르게 분포하는 최적의

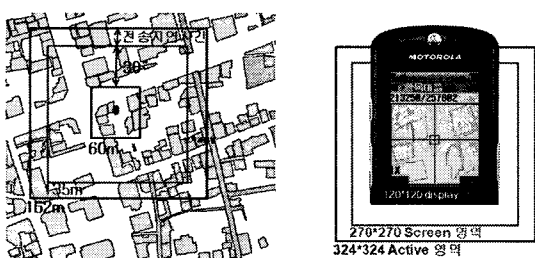


그림 3. 스크린 영역과 액티브 영역 추출

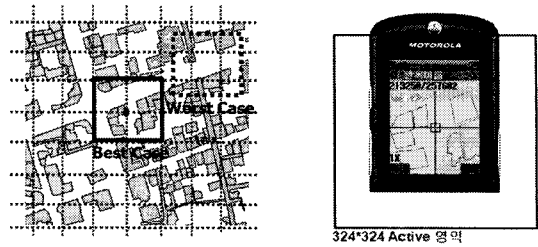


그림 4. 서버의 맵 분할

크기이며, 전송 시간을 고르게 분포하도록 한다. 맵 분할 방법은 하나의 영역 경계값이 결정되면 맵 전체의 경계값에서 하나의 영역 경계값으로 나누고 객체의 중심 좌표가 영역 경계값의 포함 여부를 판단하여 분할한다. 그림 4는 서버의 맵 분할 방법을 나타낸다.

3.2 중복성 제거를 위한 맵 캐시 메커니즘

본 논문에서는 중복성을 제거하기 위해 서버에서 맵 분할 시 각 영역별로 ID를 두고 관리한다. 그림 5와 같이 모바일 디바이스에서 스크린 영역을 벗어나 맵 재전송을 요청하면, 현재 모바일 디바이스가 보유하고 있는 영역의 ID와 중복성을 체크 후 중복되지 않는 부분들에 대해서만 전송한다. 모바일 디바이스에서는 새롭게 받아온 영역들과 기존에 존재하던 영역들을 조합하고, 재영역 분할을 통해 액티브 영역에 포함되지 않는 영역에 대해서는 캐시에 저장해 놓는다. 캐시의 크기는 좌우전후 각각 한 번씩의 재전송을 보완할 수 있는 크기로 설정한다.

3.3 적응적 맵 분할 메커니즘

적응적 맵 분할은 맵 전송시간 지연을 방지하기 위한 기법이다. 서버의 맵 분할시 데이터의 크기별로 분할하지 않고, 특정 영역으로 분할하기 때문에 그림 6과 같이 객체들이 밀집된 지역과 그렇지 않은 지역

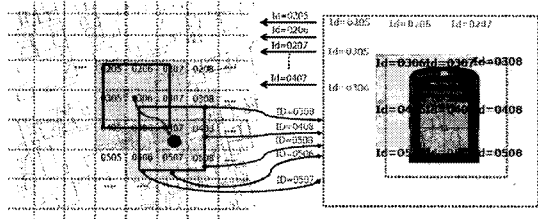


그림 5. 중복성을 제거한 맵 캐싱 기법

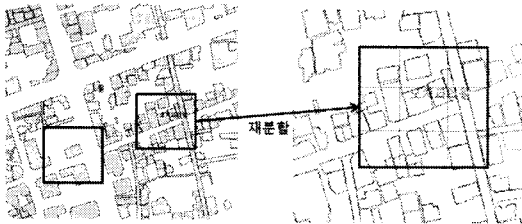


그림 6. 적응적 맵 분할 메커니즘

으로 구분되어 질 수 밖에 없다. 이는 데이터 전송에 있어서 전송 시간이 일정치 않을 수 있을 뿐만 아니라, 객체들이 매우 밀집된 변화가에 대해서는 전송 시간이 너무 지연되는 경우가 있을 수 있다. 본 논문에서는 이를 방지하기 위해 밀집된 지역에 대해서는 재분할을 수행하여 전송하도록 한다. 재분할 기준은 맵 전송시간을 기준으로 요청시간과 전송 완료시간을 비교하여 평균치를 내고 이 평균치를 크게 벗어날 경우에 서버에 전송 요청하는 영역에 대해 재분할하여 전송하도록 요청한다. 이에 서버는 포함되는 영역의 분할 기준을 한 단계 높인, 기존 액티브 영역 대비 1/4 크기에서 1/9 크기로 좀 더 세분화하여 분할하고 전송한다.

4. GridGML 설계 및 구현

GridGML은 본 논문에서 제안한 분할 및 캐싱 기법을 적용하고 이를 평가 및 활용할 수 있도록 설계, 구현한다. GridGML은 크게 서버 부분과 모바일 디바이스 부분으로 나눈다. 서버는 GML 가시화 및 모바일 디바이스로부터 맵 요청이 있을 경우 액티브 영역을 포함하는 분할 영역을 전송한다. 모바일 디바이스는 GPS로부터 수신된 WGS84 좌표를 TM 좌표로 변환 후 서버에 맵을 요청하며 서버로부터 전송된 맵에 대해서는 다시 한 번 분할하여 가시화 한다. 또한, 모바일로 전송된 맵 데이터는 ID를 부여하여 관리한다. GridGML은 구현 환경과 테스트를 위해 실제 모바일 환경에서의 UDP 통신을 대신해 TCP/IP 통신으로 설계한다. 좌표변환 모듈은 실시간 시뮬레이터가 가능하도록 모바일 디바이스에 적용한다. 그림 7은 GridGML의 전체 구조이다.

4.1 맵 서버

맵 서버는 GML 문서를 분석하여 속성 추출을 담

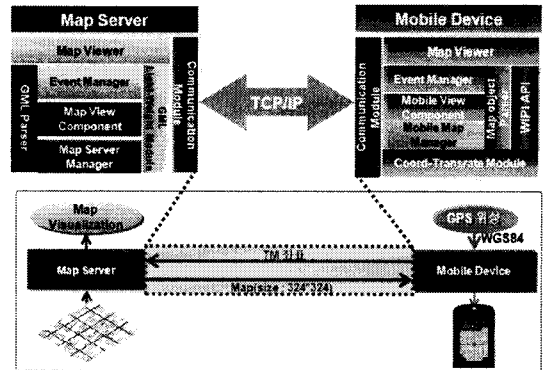


그림 7. GridGML의 구조

당하는 GML 파서와 추출된 데이터를 통해 맵을 가시화하는 맵 가시화 컴포넌트, 맵의 이벤트 처리를 위한 Event Manager, 맵을 분할 관리 및 모바일 디바이스로의 전송을 담당하는 Map Server Manager, 맵 전송시 GML 경량화를 위한 GML Light-weight Module로 구성된다.

4.1.1 GML 파서

GML 파서는 XML 파서를 통해 GML 파싱을 유도한다. XML 파서는 SAX 방식으로 메모리 소비와 CPU 자원의 활용도를 높인다. GML 파서는 SAX에서 제공하는 표준 라이브러리 모듈 중 XMLParser 클래스를 상속받아 GMLParser 클래스를 정의한다. GMLParser 클래스는 GML Schema 기반으로 각 엘리먼트들을 정의하고, 각 엘리먼트들을 처리하기 위한 이벤트 형태로 정의한 핸들러를 호출한다.

GMLParser 클래스는 엘리먼트를 키/값 형식으로 태그를 처리한다. 키는 태그 이름[<, >, 생략]이고, 값은 시작 태그와 종료 태그를 실행할 함수 이름이며 튜플로 구성된다. 예를 들어, <gml id="2" type="simple"> 태그일 경우, GMLParser 클래스는 gml_start_tag[{'id':'2', 'type':'simple'}]을 호출한다. 함수 호출 방식은 시작 태그의 경우에는 handle_starttag 함수를 호출하며 종료태그의 경우 handle_endtag 함수를 재정의 하면서 GML의 루트 엘리먼트에서 파일의 끝까지 반복한다. 파싱이 종료되면 맵 가시화 및 모바일 전송을 위해 Object형태로 저장하게 된다. GML_Object는 객체의 Geometry Type, 객체의 중심좌표, 그리기 위한 좌표들로 구성되어 저장된다. 그림 8은 GML 파싱 과정이다.

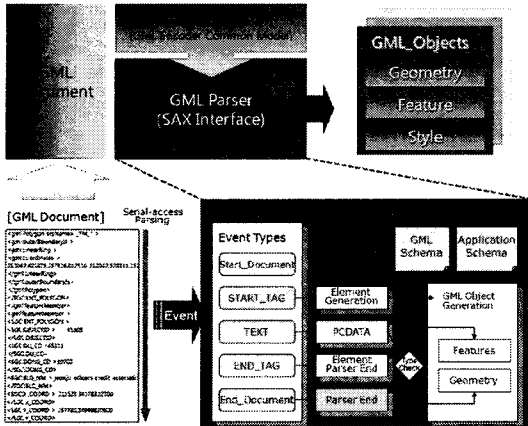


그림 8. GML Parser 구조

4.1.2 맵 가시화 컴포넌트

맵 가시화 모듈은 GML 파서에서 생성된 GML_Object의 맵 속성들을 가지고 Viewer에 가시화 하며, 유저 이벤트에 대한 처리도 담당한다. 먼저 맵 가시화를 위해 GML이 표현 가능한 Geometry Type들에 대해 각 Type별 Draw()라는 이미지 프로세싱 메소드들을 미리 구현하고 Visualization Component라는 그룹으로 패키지화한다. 맵 가시화는 GML_Object의 Type속성을 가지고 Visualization Component에 접근해 해당 Type에 매칭되는 Draw() 메소드를 호출한다. 호출된 Draw() 메소드는 가시화 과정에서 가시화시 사용자로부터 건물과 도로의 식별성을 두기 위해 각 Color 설정값을 갖고 있는 Color Manager에서 Color를 설정한다. Color 설정이 끝나면 실제 Viewer에 가시화를 위해 GML_Object의 Coordinates와 Text를 가져와 가시화 작업을 수행한다. 또한, 가시화 Module은 User Event에 대한 처리를 위해 Event Manager를 두고 이벤트가 발생되면 해당 이벤트를 분석하여 원하는 이벤트에 맞는 X, Y Boundary값을 설정하고 reDraw() 메소드를 호출해 가시화 루틴을 다시 한 번 수행한다. 그림 9는 맵 가시화 과정이다.

4.1.3 맵 서버 관리자

맵 서버 관리자(Map Server Manager)는 GML을 분할 관리하고 맵 요청에 대해 특정 분할영역을 전송하는 역할을 하며 수행과정은 그림 10과 같다.

모바일 디바이스에서 좌표, 액티브 영역과 오브젝

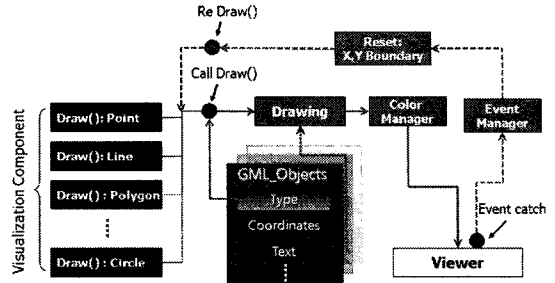


그림 9. 맵 가시화 과정

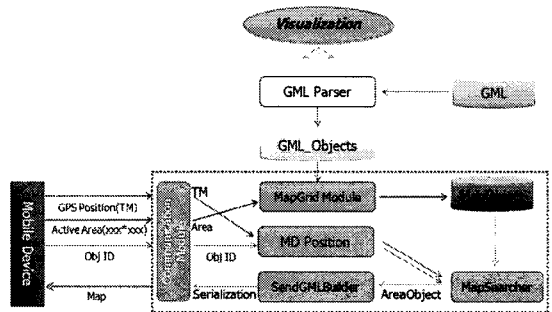


그림 10. 맵 서버 관리자의 수행 과정

트 ID를 전송하면 이를 커뮤니케이션 모듈에서 수신하고 좌표와 오브젝트 ID는 MD Position으로, 액티브 영역은 GridGML Module에 전송한다. 액티브 영역을 전송받은 MapGrid Module은 GML 가시화에 사용된 GML_Object에 접근하여 액티브 영역 크기별로 맵을 분할하고 MapObject로 저장한다. 모바일 디바이스의 좌표와 보유 오브젝트 ID를 전송받은 MD Position은 MapSearcher에 보내 MapObject로부터 재전송 영역 추출 및 중복성 제거를 수행하고 이를 SendGMLBuilder에서 바이트화하여 모바일 디바이스로 전송한다.

4.2 모바일 디바이스

모바일 디바이스는 GPS 좌표를 수신하여 서버에 맵을 요청한다. 또한 서버로부터 수신된 맵은 모바일 디바이스의 디스플레이 영역을 기준으로 재분할하여 가시화 한다.

4.2.1 모바일 가시화 컴포넌트

모바일 가시화 컴포넌트는 WIPI 플랫폼을 기반으로 한다. Map Object Parser에 의해 서버로부터 전송

받은 Map Object를 분석하고, GML Feature의 객체들을 Viewer에 가시화 한다. Map Object Parser에 의해 Map Object의 GML_Feature 내용들을 추출해 오면 Geometry Type에 따라 Visualization Component에 접근하여 해당 객체의 Type에 맞는 Draw() 메소드를 호출하고 다시 GML_Feature의 Coordinates를 가져와 도로와 건물간의 식별성을 위한 가시화 색깔을 설정하고 Drawing 과정을 수행하여 최종적으로 Viewer에 가시화 한다. 모바일 디바이스의 이벤트에 대해서는 Event Manager에 의해 이벤트를 추출하고 해당 이벤트에 따라 X,Y Boundary값을 재설정하여 재 가시화를 하면, 재설정된 Boundary값에 따라 초기 가시화 루틴을 재 수행한다. 또한 확대/축소 이벤트에 대해서는 모바일 디바이스의 4개의 가시영역이 변화하는데, 이에 따라 새로운 4개의 가시영역을 재설정하고, 이를 Mobile Manager에 전달하여 서버의 맵 분할 영역에 대해서도 재설정한다. 그림 11은 모바일 디바이스의 맵 가시화 과정이다.

4.2.2 모바일 관리자

모바일 관리자(Mobile Manager)는 서버에 맵 요청 및 수신을 담당한다. 모바일 관리자는 전송받은 맵을 4개의 가시영역별로 분할하며, 스크린 영역을 벗어날 경우 서버에 맵 재전송을 요청한다. 모바일 관리자의 작업 수행과정은 맵 서버로부터 맵을 전송받으면 바이트 형식으로 되어있는 Map Object를 ReceiveMapObject에서 복원하여 재저장한다. Grid Area는 현재 보유중인 Map Object의 영역별로 분할하기 위해 AreaState로 보내고 AreaState는 맵의 현재 영역을 구분지어 각각의 수행 루틴으로 전달한다. 스크린 영역을 벗어나 액티브 영역에 진입했을 때

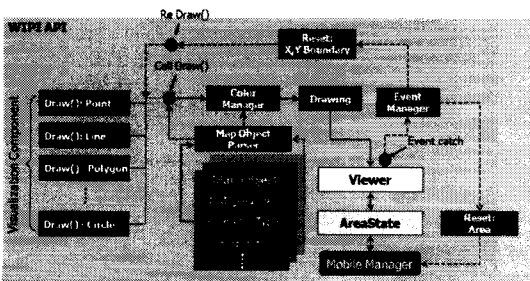


그림 11. 모바일 디바이스의 맵 가시화 과정

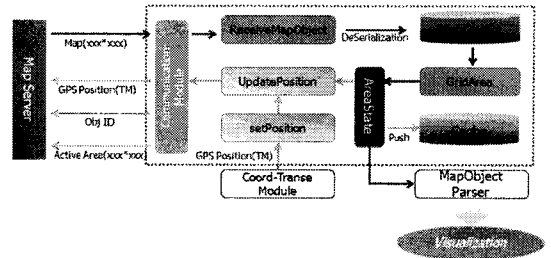


그림 12. 모바일 관리자의 수행 과정

에는 Update Position을 호출하여 현재 좌표를 GPS로부터 수신하여 모바일 디바이스에 맵 재전송 요청을 수행한다. 그림 12는 모바일 관리자의 수행 과정이다.

5. 가시화 및 성능평가

5.1 GridGML 가시화

그림 13은 GridGML의 맵 서버와 모바일 디바이스로 맵을 공급하여 가시화한 화면이다. 먼저 맵 서버는 도로와 건물은 색상으로 구분한다. 인터페이스는 맵 가시화창을 중심으로 위쪽엔 메뉴바(①)가 있고, 맵 가시화창을 기반으로 모니터링 아이콘(②)이 가시화되며, 하단에는 이벤트 아이콘(③)이 존재한다. 그리고 모바일 디바이스는 맵 서버로부터 맵을 실시간 공급 받고 이를 Aroma WIPI Emulator에 가시화 하는 화면(④)이다[13,14]. 이 경로는 “모 치과의원”에서부터 “전주 동부신협” 옆 골목까지 이동한 화면이다.

5.2 성능평가

동일한 이동경로에서 1Km의 범위를 벗어나는 데에 대해 본 논문에서 제안한 기법들의 성능을 평가한

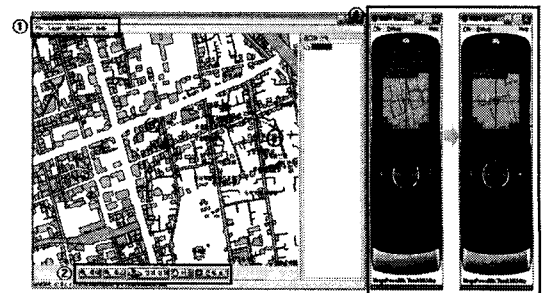


그림 13. GridGML 맵 가시화 및 모바일 디바이스 가시화

다. 먼저 동적 맵 분할 및 전송 기법의 성능을 분석하고, 적응적 맵 분할 기법의 성능을 평가한다. 성능평가를 위한 가상 환경으로 100Mbps 대역폭의 동일 네트워크상에 연결되어 있는 2.8GHz의 CPU와 1GB 메모리를 구성하고 있는 동일 데스크톱 PC를 통해 서버와 모바일 디바이스로 구축하고 모바일 디바이스를 대신할 Aroma WPI Emulator v1.1.1.8을 사용하여 성능을 평가한다.

5.2.1 동적 맵 분할 및 전송 기법에 대한 성능평가

그림 14는 동적 맵 분할 및 전송 기법과 단순 맵 전송 기법을 비교한 그래프이다. 단순 맵 전송 기법은 맵을 분할 관리하지 않고 모바일 디바이스의 요청시 매번 일정한 영역을 추출하여 전송하는 기법으로써, 동적 맵 분할 및 전송 기법에 비해 초기에 보내는 데이터 크기는 작으나 그 후 이동시에는 데이터 크기가 크다. 따라서 그림 15와 같이 동적 맵 분할 및 전송 기법은 데이터를 전송하고 처리하는데 걸리는 시간이 감소한다.

모바일 디바이스에 누적되는 데이터 크기는 6번째 전송이 이루어지는 720m 지점에서 그림 16과 같이 단순 맵 전송 기법이 동적 맵 분할 및 전송 기법에 비해 증가하는 것으로 결과를 보인다.

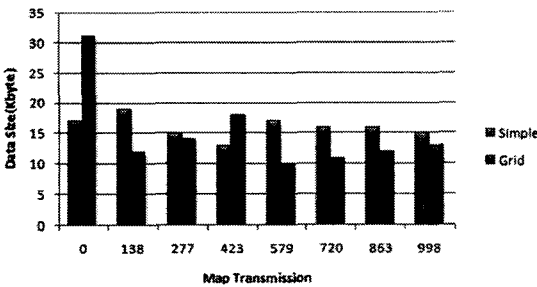


그림 14. 맵 재전송에 따르는 데이터 크기

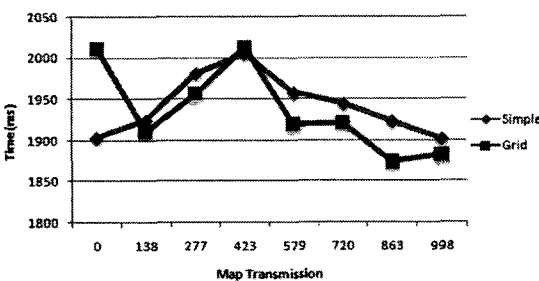


그림 15. 맵 재전송에 따르는 데이터 전송 및 처리 시간

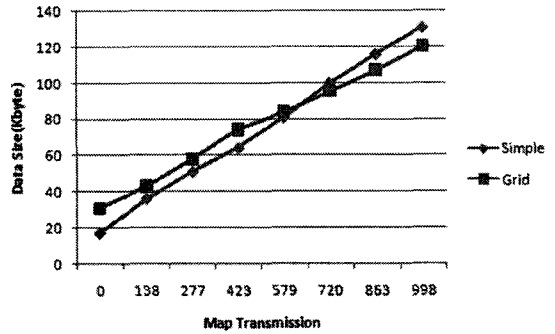


그림 16. 모바일 디바이스에 누적되는 데이터 크기

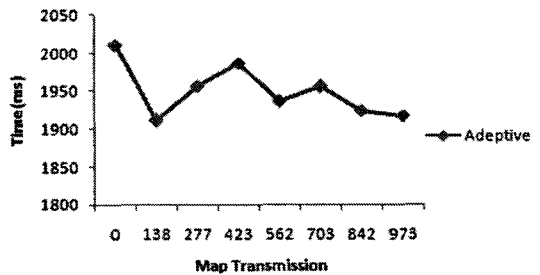
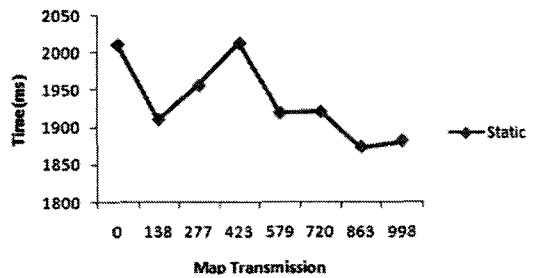


그림 17. 적응적 맵 전송 기법 성능 평가

5.2.2 적응적 맵 분할 기법

그림 17은 고정된 맵 분할 영역을 있는 그대로 전송하는 고정 맵 분할 기법과 밀집 지역 등에 대해서는 적응적으로 맵을 재분할하여 전송하는 적응적 맵 분할 기법과의 비교이다. 고정 맵 분할 기법은 전송 시간이 큰 폭으로 변화하고 있으나, 적응적 맵 분할 기법에서는 대체적으로 1900ms와 2000ms사이에서 일정한 전송 시간을 보이고 있다.

6. 결론 및 향후 연구과제

본 논문에서는 모바일 서비스에 취약한 GML을 이용하여 본 논문에서 제안한 지도의 분할 및 캐싱 기법

을 적용하여 실시간 맵 전송 및 모바일 맵 가시화를 하는 GridGML을 구현하였다. GridGML은 현재 대다수의 모바일 LBS의 기반 데이터인 맵 데이터에 대해 비표준화에 따르는 중복 구축 등의 경제적 손실을 방지하기 위해 GML 맵 포맷을 수용하였다. 그러나 GML을 그대로 사용하여 모바일에 적용하기란 ASCII 코드의 특성상 파일의 가중성에 의해 성능적 제약에 부딪히게 된다. 이에 GridGML은 GML 데이터의 맵 가시화 속성들만을 추출하여 Map Object라는 클래스 객체로 저장한다. Map Object는 GML 데이터 경량화를 위해 offset 값을 적용하여 GML 데이터의 가장 많은 부분을 차지하는 영역좌표들에 대해 offset 값을 빼고 저장함으로써 데이터 크기를 1/4로 줄이는 효과를 얻었다. 또한 GridGML은 모바일 디바이스의 실시간 맵 가시화를 위해 모바일 디바이스의 디스플레이창을 기준으로 디스플레이 영역, 스크린 영역, 액티브 영역, 캐쉬 영역 총 4개의 영역으로 분할하여 가시화한다. 맵 서버는 모바일 디바이스의 액티브 영역에 따라 액티브 영역 대비 1/4 크기로 맵을 동적으로 분할하여 관리하여 맵 전송의 효율성을 보였다. 또한 맵을 분할 때 각 영역마다 ID를 부여하여 모바일 디바이스로부터 맵 요청시 모바일 디바이스가 보유중인 영역을 뺀 나머지 부분만을 전송함으로써 데이터의 중복성을 고려한 맵 캐쉬 메커니즘을 제안하였다. 특히 맵의 밀집 지역은 재분할하여 전송하여 맵 전송시간을 일정하게 유지하도록 하였다.

향후 연구 과제로는 GML 데이터의 좀 더 효율적 전송 및 관리를 위해 ZIP 알고리즘 등 압축 알고리즘을 적용하여 좀 더 경량화 시키고, preCache 기법을 적용하여 방향성을 고려한 예측 맵 전송 메커니즘을 연구한다. 또한 최근 활발하게 연구되고 있는 맵의 3차원 가시화에 대해서도 LOD를 적용하여 연구함으로써 실질적인 지리 표현이 가능하도록 한다.

참 고 문 헌

- [1] OpenGIS Consortium, Inc., "Geography Markup Language(GML) Implementation Specification," <http://www.opengeospatial.org/docs/02-023r4.pdf>
- [2] OpenGIS Consortium, Inc., "OpenGIS Location Service Core Services," <http://www.opengeo.org/>
- [3] Z. Guo, S. Zhou, Z. Xu, and A. Zhou, "G2ST: a novel method to transform GML to SVG," *Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems*, pp. 161-168, 2003.
- [4] S. Shekhar, R. R Vatsavai, N. Sahay, T. E. Burk, and S. Lime, "WMS and GML based interoperable web mapping system," *GIS: Geographic Information Systems*, pp.106-111, 2001.
- [5] S. Takino, "GIS on the fly to realize wireless GIS network by Java mobile phone," *Proceedings of the Second International Conference on Web Information Systems Engineering*, Vol.2, pp. 76-81, 2002.
- [6] Ron Lake, David S. Burggraf, Milan Trninic, and Laurie Rae, *Geography Markup Language (GML) : Foundation for the Geo-Web*, John Wiley & Sons Inc, 2004.
- [7] 이광엽, 이진석, 송은하, 정영식, "GMLGen: WIPI 기반 모바일 GIS 가시화," 한국정보처리학회논문지, 제12권 제1호, pp. 1469-1472, 2005.
- [8] 이광엽, "모바일 GIS 가시화를 위한 GML Maker 개발," 원광대학교 정보컴퓨터교육학과 석사학위논문, 2005.
- [9] 삼성전자 통신연구소, "LBS 측위기술 개발," LBS 산업협의회 창립 총회 및 기념 세미나, 2003.
- [10] J. Green, D. Betti and J. Davison, "Mobile Location Services: Market Strategies," Ovum Ltd, 2000
- [11] 한국전자통신연구원, "LBS 기술/시장보고서," pp.11-17. 2002.
- [12] 최혜옥, "무선인터넷 위치서비스를 위한 LBS 플랫폼," 전파 제107호(무선관리단), 2002.
- [13] Wireless Internet Standardization Forum, "Mobile Standard Platform WIPI 2.0.1," <http://wipi.or.kr>, 2004.
- [14] (주)아로마소프트, "AROMA-WIPI Emulator Evaluation Ver.1.1.1.8," <http://www.aromasoft.com/>



송 은 하

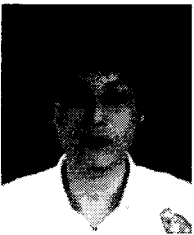
1997년 원광대학교 통계학과 학사
2000년 원광대학교 컴퓨터공학과 석사
2006년 원광대학교 컴퓨터공학과 박사

2007년~현재 원광대학교 전기전자 및 정보공학부 전임 강사
관심분야 : 그리드컴퓨팅, 시맨틱그리드, 분산병렬시스템, LBS



한 원 희

2007년 원광대학교 전기전자 및 정보공학부 학사
2007년~현재 원광대학교 컴퓨터공학과 석사과정
관심분야 : GIS, LBS, USN



박 용 진

2006년 원광대학교 전기전자 및 정보공학부 학사
2008년 원광대학교 컴퓨터공학과 석사
관심분야 : LBS, GIS



정 영 식

1987년 고려대학교 수학과 학사
1989년 고려대학교 전산학과 석사
1993년 고려대학교 전산학과 박사
2004년 웨인 주립대학교 객원 교수

1993년~현재 원광대학교 전기전자 및 정보공학부 교수
관심분야 : 그리드컴퓨팅, 시맨틱그리드, 분산병렬시스템, LBS