

무선 단말기에서 지도 서비스를 위한 시스템의 설계 및 구현

이 재 호*, 임 덕 성**, 홍 봉 희**

*한국전자통신연구원 **부산대학교 컴퓨터공학과

The Design and Implementation of System for providing mobile map service in Wireless Devices

Jai-Ho Lee, Duk-Sung Lim, Bong-Hee Hong

ETRI* Dept. of Computer Engineering, Pusan National University

요 약

무선 네트워크를 통해 공간 데이터를 요청하고 수신하는 PDA에서 지도 검색을 하기 위해 캐쉬와 공간 색인을 사용하는 환경을 대상으로 한다. 서버로부터 전송되는 공간 객체들은 캐쉬된 영역의 공간 객체와 중복이 발생하므로 이에 대한 처리 방법이 필요하다. 또한 추가되는 영역의 공간 데이터를 색인에 추가하는 연산은 Bulk-Insertion 기법이 효과적이지만 추가영역에 대한 색인 구축 연산이 무선 단말기에 부하를 준다. 그리고 교체로 인한 캐쉬 영역 삭제 시에도 신속한 영역 단위의 삭제 방법이 필요하다. 이 논문에서는 먼저 중복 객체에 대한 처리 방법을 분류하고 Bulk-Operation을 통해 캐쉬 영역의 빠른 색인 삽입, 삭제 연산을 수행하는 지도 서비스를 위한 시스템을 제시한다. 무선 단말기에 집중된 부하를 분산시키고 성능 향상을 위해 에이전트를 이용한 시스템을 설계하고 구현한다. 구현하여 실험한 결과 클리핑 방법이 효과적이며 전체적인 성능 향상을 보인다.

1. 서론

최근 PDA와 같은 무선 단말기의 보급이 증가함에 따라 무선 이동 통신망 환경에서 공간 정보를 기반으로 하는 다양한 공간 응용 서비스들이 개발되고 있다. LBS, ITS, 실시간 지도 정보 서비스 등의 응용을 위해서는 무선 단말기에서 공간 데이터 검색이 필수적이다.

PDA와 같은 무선 단말기가 무선 통신을 통한 지도 데이터를 송, 수신하는 환경을 대상으로 한다. GIS 서버는 질의 처리를 수행하고, PDA는 영역 단위의 지도 요청을 한다. 이와 같은 환경에서 무선 통신의 비용 감소와 질의 응답시간 향상을 위해 캐쉬가 필요하다. 그러나 PDA에서의 캐쉬는 저장 공간의 제약성을 가지므로 검색하는 영역만을 저장해야 하고, 효과적으로 공간 데이터를 검색하기 위해서는 색인 구성이 필요하다. 이 논문에서는 R-Tree 공간 색인을 기반으로 한다.

PDA에서 효과적으로 공간 데이터를 검색하기 위해서는 캐쉬와 공간 색인이 필요하지만 다음과 같은 문제점들이 존재한다.

첫째, 캐쉬 영역 사이의 중복 객체에 대한 처리이다. 사용자 요청 데이터만을 저장하기 위해서 질의 영역을 캐쉬의 단위로 사용해야 한다. 그래서 캐쉬된 영역과 캐쉬 될 질의 결과 영역 사이에는 중복된 객체가 존재할 수 있다.

둘째, 고 비용의 색인 구성 연산이 발생한다. 서버로 받아온 질의 영역내의 객체들을 적은 비용으로 빠르게 공간 색인에 삽입하기 위해 Bulk-Insertion 방법[1]을 적용한다. Bulk-Insertion 방법은 색인 구성과 기존의 색인에 추가하는 연산으로 이루어져

있다. 서버로부터 받아온 영역에 대한 색인을 구성하는 방법으로는 객체를 하나씩 삽입하는 방법과 Bulk-Loading[3]의 방법을 들 수 있지만 고 비용의 복잡한 연산이기 때문에 PDA에서는 큰 연산 비용을 가진다.

셋째, R-Tree 공간 색인에서 캐쉬 영역을 삭제 시 고 비용의 연산이 발생한다. 저장 공간의 부족으로 교체 발생 시에 교체 대상 영역 내에 속한 공간 객체들의 삭제 연산을 수행하게 된다. 교체 대상의 캐쉬 영역 내에는 많은 공간 객체들이 존재할 수 있고 이 객체들을 하나씩 삭제할 하게 되면 큰 연산의 부하를 초래하게 된다. 따라서 영역 단위의 빠른 삭제 방법이 필요하다.

이 논문에서는 중복객체를 처리하기 위한 방법을 분류한다. 그리고 중복 객체 처리 비용과 색인 구성 비용을 줄이기 위한 에이전트를 기반으로 한 지도 서비스 시스템 구조를 제시한다.

2. 관련 연구

사용자가 요청한 영역 중 저장되어 있지 않은 영역만을 서버로 요청하고 저장하기 위해서는 시멘틱 캐쉬[2]를 이용한 질의 영역 단위로 캐쉬를 해야 한다. 시멘틱 캐쉬는 클라이언트의 질의에 대한 결과를 단위로 하여 객체 단위에 비해 관리 비용이 적고 페이지 단위에 비해 캐쉬 단위 내의 데이터는 의미적으로 연결성을 가질 수 있다.

캐쉬된 공간 데이터의 효과적인 처리 및 검색을 위해서는 공간 색인을 필요로 한다. 공간 색인에 대한 많은 연구가 있었다. 그 중에서 R-Tree 공간 색인을 PDA의 공간 데이터에 대한 공간 색인으로 적용한다. 왜냐하면 첫째, 사용자의 요청에 의해 저장 영역이 동적으로 변화하게 되는데 이때 해쉬 기반의

공간 색인 구조 보다는 트리 구조를 가지고 있기 때문에 동적으로 확장이 용이하다. 둘째, Bulk-Insertion기법[1]을 통해 객체 단위로 하나씩 삽입하는 것이 아니라 영역 단위의 데이터의 삽입을 할 수 있다. 셋째, 영역 질의에 효과적이다.

3. 중복 객체 처리

중복 객체를 처리 하기 위한 방법들은 아래와 같다.

첫째, 질의 영역 사이의 중복 객체를 허용하여 동일한 객체를 중복 저장하는 방법이다.

둘째, 공간 객체의 영역 중 질의 영역 외의 부분을 잘라 질의 영역에 속한 부분만을 저장하는 클리핑 방법이다.

셋째, 새롭게 추가되는 캐쉬 영역에서 이미 저장된 객체를 제거하는 단일 저장 방법이다.

위에서 언급한 중복 객체에 대한 처리로 인하여 모바일 클라이언트에서 처리할 연산의 부하는 더 가중되었다. 특히 중복 저장의 경우 작은 저장 공간을 가진 PDA에서 사용하기에는 적절하지 못하다. 또한, 단일 저장의 경우 교체 시 영역간에 교차하고 있는 객체에 대한 처리 문제가 복잡하다. 따라서 이 논문에서는 모바일 환경에서 공간 객체의 효율적 검색 및 관리를 위해 클리핑 방법을 적용한다.

4. 에이전트를 이용한 시스템 구조

이 논문에서 제시하는 모바일 환경에서 지도 데이터의 효율적 검색을 지원하는 전체 시스템의 구조는 그림 1과 같이 GIS 서버, 에이전트, 모바일 클라이언트로 구성된다.

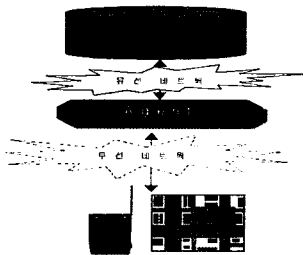


그림 1 시스템 구조

에이전트와 서버는 유선 네트워크를 통해 연결 되어 있고 에이전트는 모바일 클라이언트와 무선 네트워크를 통해 통신한다. 처리 속도가 상대적으로 느린 모바일 클라이언트의 부하와 무선 네트워크를 통한 통신 양을 줄이기 위한 구조이다.

4.1 에이전트

4.1.1 에이전트의 정의 및 기능

이 논문에서의 에이전트는 다음과 같이 정의 한다.

“GIS 서버로부터 공간 데이터를 가져오기 위한 질의를 생성하고 질의의 결과로 가져온 공간 데이터에 대해 PDA에 이미 캐쉬된 공간 객체와 중복된 객체에 대한 처리 연산과 공간 색인을 구성 하는 소프트웨어” 라고 정의한다.

에이전트의 기능은 다음과 같다.

첫째, 서버로 질의할 질의 영역을 생성한다.

둘째, 질의 결과에 대해 중복 객체를 처리하기 위한 연산으로 클리핑 연산을 수행하거나 중복된 객체를 제거한다.

셋째, R-Tree 공간 색인을 구성하는 작업을 수행한다.

위와 같은 에이전트를 사용하는 목적은 모바일 클라이언트에 집중된 연산의 부하를 감소시키고 컴퓨팅 파워가 월등히 우수한 에이전트에서 처리함으로써 모바일 클라이언트의 지도 검색 속도의 향상을 위해서이다.

4.1.2 에이전트 구조

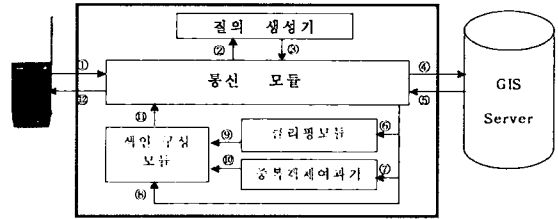


그림 2 에이전트 구조

에이전트는 그림 2와 같이 질의 생성기, 통신 모듈, 클리핑 모듈, 중복 객체 제거기, 색인 구성 모듈로 구성되어 있다. 중복 객체의 3가지 처리 방법에 대한 실험을 위해서 3가지 방법을 모두 수행할 수 있는 에이전트 구조를 설계하였다.

에이전트의 수행 시나리오는 다음과 같다.

- 단계1,2 — 모바일 클라이언트에서 사용자 요청 영역, 사용자 요청 영역과 겹치는 캐쉬된 영역 정보를 수신한다.
- 단계3 — 질의 생성기는 서버에 요청할 영역 질의를 생성하여 전달한다.
- 단계4,5 — 생성된 질의를 서버로 요청하고 결과를 받는다.
- 단계6,7,8 — 서버로부터 질의 결과를 수신한다.
- 단계9 — 질의 결과를 클리핑한다.(클리핑의 경우)
- 단계10 — 질의 결과 중 중복객체를 제거한다.(단일 저장의 경우)

4.2 모바일 클라이언트의 구조

모바일 클라이언트는 사용자 인터페이스, 질의 처리기, 캐쉬관리자, 캐쉬, 통신 모듈로 구성되어 있다. 아래 그림 3은 모바일 클라이언트의 구조를 보여준다.

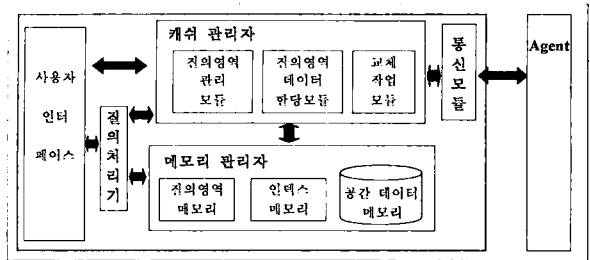


그림 3 모바일 클라이언트의 구조

- 사용자 인터페이스
 - 화면상에 지도를 디스플레이하고 사용자의 요청을 받아들이는 모듈이다. 지도의 확대, 축소, 드래깅 방법으로 panning 기능을 제공한다.
- 질의 처리기
 - R-Tree 공간 색인에 대해서 영역 질의를 수행하여 결과를 반환하는 기능을 한다.
- 통신모듈
 - 사용자가 요청한 영역 정보와 캐쉬된 영역 중 사용자가

요청한 영역과 겹치는 영역의 정보를 에이전트에게 송신하고 에이전트로부터 캐쉬되지 않은 영역의 데이터와 공간 색인 정보를 수신한다.

● 캐쉬관리자

질의 영역관리 모듈은 캐쉬된 영역 정보들을 저장하고 관리한다.

질의 영역 데이터 할당 모듈은 서버로부터 받아온 질의 영역의 데이터를 공간 데이터 메모리에 저장하고 에이전트에서 이미 구성된 색인 정보를 이용하여 기존의 공간 색인에 추가하는 역할을 한다. 이때 공간 색인에 추가하는 연산은 Bulk-Insertion 기법을 사용한다.

교체 작업 모듈은 질의 영역 데이터 할당 모듈에서 요구하는 데이터 메모리의 공간을 확보하기 위해 교체 작업을 수행한다. LRU기법을 이용하여 교체 대상을 선택한 후 교체 영역에 해당하는 정보를 삭제하고, 클러스터링되어 저장된 교체 영역에 속한 공간 객체들을 공간 데이터 메모리에서 삭제하며 인덱스 메모리에서 색인 정보를 제거한다. 색인에서 교체 영역을 제거하기 위해서 Bulk-Deletion 연산을 적용한다. 지면관계로 Bulk-Deletion 알고리즘 기술은 생략한다.

● 메모리 관리자

질의 영역 메모리는 캐쉬된 영역에 대한 정보를 저장하는 메모리 영역이다. 하나의 영역에 대해서 영역의 MBR, 최근 사용 시간, 영역의 데이터의 저장 위치 정보를 저장하고 있다.

인덱스 메모리는 캐쉬된 전체 공간 데이터의 R-Tree 색인 정보를 저장하는 메모리 영역이다.

공간 데이터 메모리는 캐쉬된 공간 데이터를 저장하는 메모리 영역이다. 캐쉬 영역 별로 클러스터링되어 저장되어 있다.

5. 구현 및 실험 평가

5.1 구현 및 실험 환경

이 논문의 구현을 위해서 모바일 클라이언트는 Palm IIIc Palm OS 3.5 ROM을 탑재한 에뮬레이터를 사용하였다. 에이전트는 Windows 2000 환경에서 사용하였으며, GIS 서버는 Linux 환경을 사용하는 파일 시스템 기반의 서버를 사용하였다. 실험 지도 데이터는 부산 지역 중 일부 영역을 대상으로 하였고, 그 중 데이터의 양이 가장 많은 4가지 레이어에 대해서 실험하였다

5.2 성능 실험

이 실험의 목적은 PDA에서 캐쉬 영역 사이의 중복 객체 처리의 방법들을 비교 분석하여 가장 효율적인 방법을 결정하기 위함이다. 각 처리 방법에 대해서 처리 시간과 저장 공간의 비교 실험을 하였다.

5.2.1 저장 공간에 대한 실험

모바일 클라이언트에서 사용자가 임의 영역에 대해서 영역 질의를 할 때 에이전트에서 각 중복 객체의 처리 방법들에 의해 생성되는 데이터의 통신량을 측정하였다.

그림 4는 두 면 레이어에 대해 실험한 결과이다. 질의 형태는 임의 영역에 대한 영역 질의로서 100회 수행 후 평균값을 나타낸다. 단일 저장하는 경우를 기준으로 중복 저장하는 경우와 클리핑할 경우의 데이터의 양의 비율을 그래프로 나타내었다. 두 면 레이어의 객체들이 단순한 기하 모양을 가지고 있어 클리핑을 했을 때 단일 저장 방법보다 데이터의 양에서 이점이 없다.

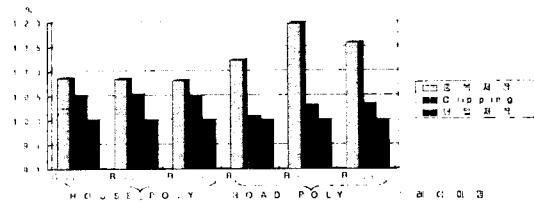


그림 4 중복 객체 처리 방법들의 데이터 양 비교면

그러나 그림 5는 두 선 레이어에 대한 실험 결과를 보면 클리핑을 했을 때 데이터의 양이 단일 저장보다 현저히 줄어든 것을 볼 수 있다. 중복 저장은 다른 두 방법에 비해 많은 저장 공간을 요구한다는 것을 알 수 있다.

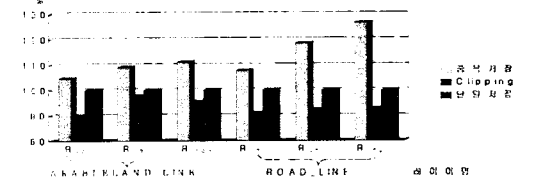


그림 5 중복 객체 처리 방법들의 데이터 양 비교면

5.2.2 처리에 시간에 대한 실험

위의 실험과 병행하여 각 처리 방법들의 시간을 측정하였다. 각 처리 방법들의 시간을 비교함으로써 처리 부하를 알아 보기 위한 실험이다. 실험결과는 3가지 방법 모두 매우 미미한 차이를 보였으며 처리속도가 우수한 에이전트에서 중복 객체를 처리함으로써 중복 객체의 처리 연산은 전체 시스템에 거의 부하를 주지 않는다는 것을 보였다.

5.2.3 실험 결과 분석

이 실험에서 단순 공간객체가 아닌 복잡한 공간객체의 검색의 경우 클리핑 방법을 사용하여 중복 객체를 처리할 경우에 다른 방법에 비해 데이터의 양이 많이 줄어든다는 것을 알 수 있다. 따라서 통신 비용감소 효과가 있고 모바일 클라이언트의 저장공간의 활용도 좋아지게 된다. 그리고 처리 시간의 부하는 모바일 클라이언트가 아닌 에이전트가 처리함으로써 전체적인 질의 검색시간을 향상시킬 수 있다.

6. 결론

무선 통신이 가능한 모바일 클라이언트에서 지도 서비스를 하기 위한 에이전트 기반의 시스템 구조를 설계 및 구현하였다. 시멘틱 캐쉬를 적용하였고, 캐쉬된 데이터의 공간 색인에 새로운 캐쉬 영역의 빠른 삽입과 삭제 연산을 위해서 Bulk-Operation을 이용하였다. 또한 컴퓨터 파워가 낮은 모바일 클라이언트에 집중된 연산을 에이전트를 이용하여 분산 처리하기 위한 구조를 제시하고 질의 처리 성능 향상을 하였다. 또한, 중복된 객체에 대한 처리 방법을 분석하고 그에 따른 실험 평가를 하였다.

7. 참고 문헌

[1] Li Chen, Rupesh Choubey, Elke A. Rundensteiner : Bulk-insertions into R-Trees ACM-GIS 1998: 161-162
 [2] Shaul Dar, Michael J. Franklin, Bjorn T. Jonsson, Divesh Srivastava, and Michael Tan, "Semantic Data Caching and Replacement", In Proceedings of the 22nd Very Large Data Bases, 1996
 [3] Van den Berchken, Seeger, Widmayer: "A General Approach to Bulk Loading Multidimensional Index Structures", Int Conf. on Very Large Databases, VLDB 1997