

지도 일반화를 지원하는 계층화된 공간 색인 기법†

Leveled Spatial Indexing Technique supporting Map Generalization

이기정*, 황보택근**, 양영규***

Ki-Jung Lee, Taeg-Keun Whangbo, Young-Kyu Yang

요약 핸드폰의 화면 크기 때문에 핸드폰에서의 지도 서비스는 문제점을 가지고 있다. 핸드폰과 같은 작은 화면에서 효율적으로 지도 데이터를 표현하기 위해서는 첫째, 지도 일반화를 이용하여 상세한 지도 데이터를 간략하게 만드는 과정이 필요하며, 둘째, 간략화된 데이터를 계층화하여 사용자의 확대·축소 명령을 빠르게 수행할 수 있어야 한다. 그러나, 기존의 연구들은 일부의 지도 일반화만 지원하거나 실시간으로 처리하지 못하는 문제점들을 가지고 있다.

본 논문에서 제안하는 LMG-tree는 계층화된 하나의 색인 트리를 사용함으로써 저장 공간의 효율성을 가지고 있으며, 지도 일반화를 지원하여 실시간 지도 서비스가 가능하도록 하였다.

Abstract Map services for cellular phone have problem for implementation, which are the limitation of a screen size. To effectively represent map data on screen of cellular phone, it need a process which translate a detailed map data into less detailed data using map generalization, and it should manipulate zoom in·out quickly by leveling the generalized data. However, current spatial indexing methods supporting map generalization do not support all map generalization operations.

In this paper, We propose a leveled spatial indexing method, LMG-tree, supporting map generalization and presents the results of performance evaluation.

주요어 : 지도일반화, 공간색인, 모바일, 지리정보시스템

Keywords : Map Generalization, Spatial Indexing, Mobile, GIS

1. 서론

무선 이동통신의 성장과 모바일 콘텐츠의 다양성이 증가하면서 사용자들은 휴대폰에서 유선 인터넷 서비스의 품질과 동일한 수준의 정보 서비스들을 요구하고 있다. 그 중 지도 서비스는 휴대폰의 이동성이라는 특성 때문에 성장의 잠재력이 높은 서비스로 대두되고 있다.

핸드폰에서의 지도 서비스는 핸드폰의 작은 메모리, 낮은 연산력, 그리고 작은 화면 크기로 인하여 효율적으로 정보를 제공하기가 어려운 단점을 가지고 있다.

핸드폰에서 효율적으로 지도 데이터를 표현하기 위해서는 첫째, 지도 일반화(Map Generalization)를 이용하여 상세한 지도 데이터를 간략하게 만드는 과정이 필요하며, 둘째, 간략화된 데이터를 계층화하여 사용자의 확대·축소 명령을 빠르게 수행할 수 있어야 한다.

지도 일반화는 상세한 데이터를 줄여서 간략한 데이터를 만드는 기능을 제공하며, 인간의 해석 능력과 분석 능력에 적당하도록 상세화된 정도를 감소시키고 객체의 밀도를 줄이는 데 그 목적이 있다. 핸드폰과 같은 작은 화면에서 지도 데이터를 표현하기 위해서는 지도 일반화를 수행하여 상세한 데이터를 간략하

† 이 논문은 2004년도 대학 IT연구센터 육성지원사업비의 지원을 받아 연구되었음

* 경원대학교 일반대학원 전자계산학과

** 경원대학교 소프트웨어 대학 부교수

*** 경원대학교 소프트웨어 대학 교수

jcm5758@ku.kyungwon.ac.kr

kwhangbo@kyungwon.ac.kr

ykyang@kyungwon.ac.kr

게 만드는 것이 필요하다[1].

핸드폰에서 사용자의 확대·축소 명령에 따라 연산을 수행하는 것은 핸드폰의 연산 처리 능력과 메모리 문제로 인하여 쉽지 않다. 그럼으로 검색한 영역에 대한 데이터를 계층화하여 저장함으로써 검색시간을 축소하는 작업이 필요하다[2][3].

기존의 공간 색인 기법들은 지도 일반화 연산자들을 지원하지 않거나, 일부의 연산자만을 지원함으로써 핸드폰과 같은 작은 화면에서의 처리가 힘든 단점을 가지고 있다.

본 논문에서는 지도 일반화를 지원하면서 계층화된 공간 인덱스 기법을 제안하였다. 이 공간 색인 기법을 LMG-tree(Levelled Tree supporting Map Generalization)로 명명하였으며, 기존의 공간 색인 기법들과 성능을 비교하였다.

본 논문은 2장에서 기존의 공간 색인 기법들에 관한 연구에 대해서 알아보고, 3장에서 제안한 공간 색인 기법에 대하여 설명한다. 4장에서는 제안한 공간 색인 기법과 기존 연구와의 성능 비교를 수행하고, 5장에서 결론을 맺는다.

2. 관련 연구

대부분의 공간 색인 기법들은 특정 영역 안에서 지리 객체들의 효율적인 검색에 초점을 맞추어왔지만, 기존의 공간 색인에 관한 연구들은 기하 객체들이 임의의 크기로 변하거나 상호작용하는 분야에서는 적절하지 못하며, 계층화된 지리 정보를 표현하는 것은 어렵다[2][10][11][12]. 사용자가 지도를 확대하였을 경우 지도안에 표현되는 객체의 크기만 확대되는 것이 아니라 객체의 정보도 보다 상세하게 표현되어야 한다[2][3].

기존의 공간 색인 기법들을 이용하여 계층화된 지리 정보를 표현하는 방법은 크게 두 가지이다. 첫째, 모든 축척의 데이터를 따로 색인하여 별도로 저장하는 방법이다. 이 경우 각 축척별로 색인을 따로 가지고 있기 때문에 검색시 해당 축척의 색인만 사용하면 됨으로 검색 속도는 향상될 수 있지만, 사용하지 않는 나머지 축척에 대한 불필요한 저장 공간이 소요되는 단점을 가지고 있다. 둘째, 하나의 색인 구조에 가장 상세한 데이터를 저장하는 방법이다. 이 경우 가장 상세한 데이터를 가지고 있음으로 사용자가 요구하는 어떤 단계의 축척도 지원이 가능하다. 반면에, 해당 축

척의 데이터를 검색하기 위한 많은 연산량이 필요하게 되며, 검색 속도의 저하를 초래하게 된다[4][5].

기존의 공간 색인 기법들은 지도 일반화를 지원하기 위해서 첫 번째 방법을 주로 사용하였다. 지도 일반화를 수행한 데이터를 계층별로 저장함으로써, 계층별로 검색시 성능 향상을 보일 수 있다. 그러나, 첫 번째 방법의 경우 객체의 삽입이나 수정, 삭제시에 전체 계층의 데이터를 재구성하여야 하는 문제가 있으며, 지도 일반화의 연산자 중 선택과 단순화 연산자만을 지원하였다[1][3].

두 번째 방법의 경우 지도 일반화를 수행 후 저장하는 방법과 저장된 데이터를 일반화하여 표현하는 방법으로 나눌 수 있다. 지도 일반화를 수행 후 저장하는 방법은 저장 공간의 효율성이 매우 증가하지만, 계층별로 검색하기 위해서는 많은 연산량이 필요하고, 저장된 데이터를 일반화하는 방법은 일반화 연산자들이 제공되어야 한다.

지도 일반화를 지원하는 기법들로는 Reactive-tree[3], Priority Rectangle File[12], Multi-Scale Hilbert R-tree[2], 그리고 LR-tree[6]가 있다.

Reactive-tree, PR-file, MS Hilbert R-tree는 중요도 값(importance value)이나 우선 순위를 사용하여 계층화된 구조를 생성한다. 중요도 값은 객체에 나타나는 정보를 의미하며, 중요도 값이 높은 객체는 작은 축척과 큰 축척 모두에서 나타나게 되며, 중요도 값이 낮은 객체는 큰 축척에서만 나타나게 된다[2][3][12]. LR-tree는 각 계층을 따로 저장하는 방식을 이용하여 2단계의 인덱스 구조를 가진다. 1단계에서는 각 객체에 대한 객체 다중 표현이 나타나는 축척 값으로 구성되고, 2단계에서는 각 축척 값에 따른 객체 다중 표현을 나타내도록 구성된다.

Reactive-tree에서 실 객체는 중요도에 따라 해당 계층에 존재하며, 같은 계층의 객체는 같은 중요도 값을 갖는다[3]. Reactive-tree는 선택과 단순화 연산자만 지원하는 단점을 가지고 있으며[3][7], 각 계층에 해당되는 객체가 존재하지 않을 경우 해당 계층에는 실 객체가 존재하지 않는 문제가 발생할 수 있으며[2], 서로 다른 계층의 객체들간 겹침이 없어야 하는 단점을 가지고 있다[7].

PR-file은 R-file[13]에 기반하고 있으며, 임의의 축척에서 객체의 효율적인 검색과 저장을 위해서 연구되었다. PR-file은 우선 순위를 부여하여 계층의 개념으로 사용하였으며, 우선 순위가 높은 블록은 우선 순

위가 낮은 블록보다는 항상 일찍 발견된다는 속성을 가진다[2][6][7]. 우선 순위가 같은 객체들끼리 블록을 만들어 저장함으로써 별개의 객체로 이루어져 있을 경우 정상적으로 표현되지 않을 수도 있다. 더불어, 그리드 파일이 가지고 있는 문제점들을 가지고 있으며, 선택과 단순화 연산자만 지원한다.

MS Hilbert R-tree는 Hilbert R-tree[14]에 기반한 다중 축척 데이터를 지원하는 공간 인덱싱 기법 중 하나이다. MS Hilbert R-tree는 단순화 연산을 수행하는데 따른 검색 속도의 향상을 위해 실제 데이터 파일이 분할되어 저장된다. DP알고리즘을 수정하여 단순화 연산자를 지원하며, 선택 연산자를 지원하기 위해서는 데이터 크기에 기반하여 객체를 선택한다. 단점은 선택과 단순화 연산자만을 제공한다는 것과 그리드 파일이 갖는 문제점을 가지고 있다.

LR-tree는 선택 연산자만 이용해서 계층별로 트리를 구성하는 방법을 사용함으로써 다른 지도 일반화 연산을 수행할 수 없으며, 별개의 계층별 데이터를 가지고 있음으로 메모리의 비효율성을 가지고 있다. 또한, 계층별로 구성된 트리에 객체의 삽입과 삭제가 발생하면 전체 트리를 다시 구성해야 하는 문제점을 가지고 있다.

앞에서 기술한 공간 색인 기법들의 경우 지도 일반화의 일부 연산자만을 지원하거나 계층별로 기 구축된 색인을 사용함으로써 실시간으로 데이터의 간략화를 수행하기는 어렵다. 이에 본 논문에서는 계층화된 데이터를 하나의 색인 구조에 저장하고, 지도 일반화 연산자들을 지원함으로써 실시간으로 간략화가 가능한 색인 기법을 제안하고자 한다.

3. LMG-tree(Leveled tree supporting Map Generalization)

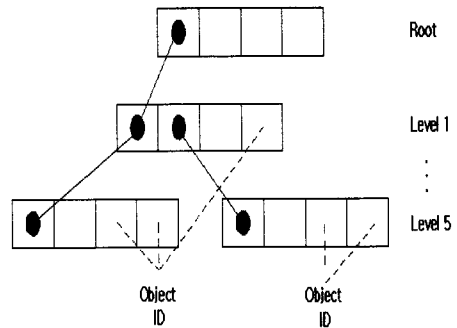
지도 일반화를 지원하는 공간 색인 기법들은 대부분 선택과 단순화 연산자만을 지원하거나 지도 데이터를 계층별로 따로 저장하여 색인하는 방법을 사용하고 있다[1][7].

LMG-tree는 하나의 트리에 전체 데이터를 저장하는 구조이다. 하나의 트리에 전체 데이터를 가지고 있음으로 삽입·수정·삭제 등의 연산 수행시에 효율적인 장점을 갖게 된다. 또한, 전체 데이터를 중요도에 따라 계층화하여 저장함으로써 검색시 해당 계층의 데

이터만 검색이 가능함으로 검색 성능의 향상을 가져올 수 있으며, 지도 일반화의 선택, 단순화, 집단화, 이동, 그리고 심벌화 연산자를 지원한다.

3.1 LMG-tree의 특징

LMG-tree는 가장 범용적으로 사용하고 있는 R-tree를 기반으로 하여[3][7], 계층화된 데이터를 저장하도록 구성되어 있으며, 색인 구조는 <그림 1>과 같다.



<그림 1> LMG-tree의 공간 인덱스 구조

<그림 1>의 검정색 원은 트리 엔트리를 의미하며, 트리 엔트리를 제외한 나머지 영역은 객체 엔트리를 나타낸다.

LMG-tree는 R-tree와 달리 루트 노드의 엔트리 수의 제한이 없다. R-tree는 루트 노드가 적어도 2개 이상의 엔트리를 가져야 한다[10]. 반면에 LMG-tree는 루트 노드의 엔트리가 1개 이상이면 된다. R-tree는 각 노드당 엔트리의 수가 2개 이상이어야 하고, 각 노드가 가질 수 있는 최대 엔트리의 수보다 작아야 한다[9]. LMG-tree는 중간 노드의 엔트리 수의 제한이 없다.

유사한 방법을 사용하는 Reactive-tree의 경우 해당 계층의 데이터가 존재하지 않는 경우 객체 엔트리가 없는 노드가 발생한다[2][3]. 그러나, LMG-tree의 경우 선택 연산자가 수행되는 동안 해당 계층의 데이터가 없으면 하위 계층의 데이터를 한 단계 올려서 중간 노드의 객체 엔트리가 항상 존재하도록 한다. 이 과정을 거침으로써 트리가 한쪽으로 치우치는 현상을 방지할 수 있다.

3.2 LMG-tree의 검색

계층별로 데이터가 저장되어 있는 공간 색인 기법에서는 계층별로 별도 구성된 R-tree에 대해 각각 검색 알고리즘을 호출하거나, 해당 계층에 대한 데이터를 추출하기 위해 검색 전 혹은 후에 별도의 계층별 데이터 정보나 검색이 필요하다[7].

LMG-tree는 검색 영역의 MBR(Minimum Bounding Rectangle)을 계산하여 검색 MBR을 포함하는 트리를 찾아서 검색을 진행한다. 검색 MBR을 포함하는 트리에서 원하는 계층 단계로 진행하여 해당 계층의 데이터를 검색한다. 이때, 계층 값들을 비교하여 원하는 계층 값보다 작은 모든 계층을 검색한다. 예를 들어, 계층 3의 영역을 검색할 경우, Root에서 시작하여 계층 1단계로 진행한다. 계층 1은 계층 3보다 계층 값이 작으므로 검색을 수행하고, 계층 2로 진행한다. 계층 2도 계층 3보다 작으므로 검색을 수행한다. 마찬가지로 방법으로 원하는 계층 값이 존재하는 계층까지 검색을 수행한다. LMG-tree의 검색 알고리즘은 다음과 같다.

1. 검색 영역(출발 노드, 목적 노드)과 탐색 계층을 입력한다.
2. 루트 노드의 엔트리들을 검색하여 검색 영역을 포함하는 엔트리를 찾는다.
3. 찾은 엔트리 하위에서 입력받은 탐색 계층으로 이동한다.
4. 해당 계층의 객체들을 검색한다.(단, 임계값이 설정되어 있으면, 설정된 임계값보다 작은 수의 객체들을 검색한다.)

3.3 LMG-tree의 삽입

객체를 삽입하고자 하는 경우에는 먼저 검색 알고리즘을 수행하여 객체 삽입 위치를 찾아서, 객체를 삽입하고, R-tree와 같이 부모 노드들의 MBR을 갱신하는 방법을 사용한다.

LR-tree와 같이 모든 계층에 대한 트리가 구성되어 있는 경우, 각 계층에 대한 트리를 모두 갱신하여야 하지만, LMG-tree와 같은 하나의 트리를 사용하는 경우 삽입 후 트리 갱신에 드는 비용이 적은 장점을 가지고 있다.

1. 검색 알고리즘을 수행하여 삽입하고자 하는 위치를 찾는다.

2. 찾은 위치에 객체를 추가한다.

3. 객체가 포함된 위치의 상위 노드의 MBR을 갱신한다.

4. 루트 노드에 도달할 때까지 트리 구조를 갱신한다.

3.4 LMG-tree의 삭제

객체의 삭제하고자 할 경우에도 먼저 검색 알고리즘을 이용하여 삭제하고자 하는 객체를 찾고, 찾은 객체를 제거한다. 객체를 제거한 후 부모 노드들의 MBR을 갱신한다. 삽입과 마찬가지로 하나의 트리로 구성된 LMG-tree는 각 계층별로 구성된 트리에서의 삭제보다 그 비용이 적게 소요된다.

1. 검색 알고리즘을 수행하여 삭제하고자 하는 객체를 찾는다.
2. 객체를 삭제한다.
3. 만약 해당 계층에 더 이상 객체 엔트리가 존재하지 않고 하위 계층에 객체가 존재하면, 하위 계층의 객체를 현재 계층으로 계층 상승시킨다.
4. 루트 노드에 도달할 때까지 트리 구조를 갱신한다.

3.5 선택(Selection)

공간 데이터베이스내의 일정 영역안의 모든 데이터를 화면에 출력하는 것은 화면의 복잡도를 증가시키고 사용자의 인지 능력을 저하시킨다. 어떤 데이터는 표현하고 어떤 데이터는 표현하지 않을 것인지 하는 문제는 매우 어려운 문제이다[9]. 본 논문에서는 같은 화면 크기라도 사용자가 인지하기 위해서 필요한 객체의 경우 중요도를 높게 선정하고, 그렇지 않은 경우 중요도를 낮게 선정하였다.

본 논문에서는 객체의 중요도를 5단계로 구분하였으며, 각 단계에서 나타나는 정보는 <표 1>과 같다.

<표 1> 단계별 정보

단계	단계별 정보
1단계	주요 도로와 회전 정보
2단계	주요 도로와 회전 정보와 주요 건물
3단계	주요 도로와 비 주요 도로와 주요 건물
4단계	도로와 건물
5단계	상세 정보

3.6 단순화(Simplification)

단순화 연산자는 선을 구성하는 점의 개수를 줄임으로써, 직선에 가까운 라인을 만들거나 건물을 단순하게 만드는 연산자이다. 본 논문에서는 도로에 관련한 객체에 대해서만 단순화를 수행하였으며, 건물은 집단화를 통해서 간략화 하였다.

본 논문에서는 회전이 발생하는 노드를 중심으로 Douglas-Peucker(DP) 알고리즘을 개선하여 사용하였다[6]. 회전이 발생하는 노드를 중심으로 단순히 직선화를 하지 않고 DP 단순화를 사용한 이유는 단순화시 단순화 정도에 대한 임계값을 두어 도로의 단순화되는 정도를 고려할 수 있어 도로의 초기 형태에 가깝게 유지할 수 있기 때문이다.

핸드폰의 작은 화면에서 효율적으로 나타내기 위해서는 일반적인 DP 단순화를 수행해서는 큰 효과를 나타낼 수 없다. 선택 연산자로 선택된 노드가 한 영역에 집중 편재되어 있는 경우 밀집된 영역에서의 사용자 인지도는 저하되게 된다. 따라서, 집중된 영역의 노드들을 삭제하거나 노드간의 거리를 증대시켜서 사용자의 인지도를 증가시킬 수 있어야 한다.

본 논문에서의 단순화는 두 가지 경우로 나뉘어진다. 첫째는 사용자가 이동하는 경로의 단순화이며, 둘째는 경로와 관련된 주요 도로들의 단순화이다. 사용자 이동 경로의 경우 출발지에서 목적지까지 이동하는 경로상의 주요 지점들을 선택 연산자를 통해서 추출하고, 추출된 주요 지점들이 너무 많은 경우가 증감될 중요한 지점들을 추가로 제거한다. 이 과정을 거침으로써 화면의 복잡성을 줄일 수 있다. 이후 단순화가 진행된 후 도로의 회전점이 회전 지역으로 나타나지 않는 경우에는 회전을 표현하기 위하여 Spline Curve를 이용하거나 단순화의 강도를 줄이는 방법을 사용한다.

주요 도로의 단순화는 검색 영역 안에 존재하는 주요 도로들에 대해서 수행된다. 주요 도로 단순화의 경우 경로와 만나는 경우와 만나지 않는 경우가 존재한다. 경로와 만나는 경우에는 경로 단순화 정도에 맞추어 주요 도로 단순화가 진행되어야 하며, 만나지 않는 경우에는 주요 도로에 대해서만 단순화한다.

3.7 집단화(Aggregation)

선택 연산자를 통해서 생성된 데이터에서 표현되지

않는 도로나 건물들의 경우 혹은 비 주요 건물들의 경우 각각의 데이터를 자세하게 표현하면 화면의 복잡성이 증가하게 되며, 화면의 복잡성을 줄이기 위해서는 비 주요 건물의 사용은 최소화하여야 한다. 집단화 과정에서는 주요하지 않거나 하나의 블록처리가 가능한 객체들의 경우 동일 블록으로 표현함으로써 화면의 복잡성을 줄일 수 있고, 표현할 데이터의 양도 줄일 수 있다.

3.8 이동(Displacement)

집단화와 단순화 연산이 수행되면 집단화와 단순화에 의해서 변경된 객체와 그 주위에 있던 객체들간의 위치가 변경되는 경우가 발생하거나, 객체들간의 겹침 현상이 발생하기도 한다[8][9]. 객체 위치의 오차나 겹침 현상을 해결하기 위해서 위치를 보정해주는 작업이 필요하며, 이 작업을 이동 연산자에서 수행한다 [1][8][9].

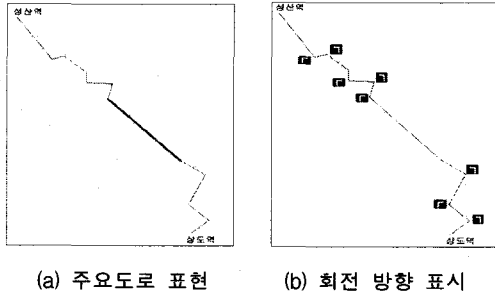
객체의 위치가 변경되는 경우를 예로 들면, 도로의 단순화 이전에 도로의 왼쪽에 위치하던 건물이 도로의 단순화 후 건물의 오른쪽에 위치하는 경우가 발생한다. 이런 경우 건물을 원래 위치로 보정하기 위해서 다음의 알고리즘을 사용한다.

1. 도로의 노드중에서 건물 P와의 거리가 가장 가까운 노드 N을 찾는다.
2. 노드 N이 속해 있는 회전 시작점 S와 끝점 E와의 거리를 계산하여 전체 거리 $dist(N_{total})$ 로 입력하고 회전 시작점과 그 노드까지의 거리 $dist(N_i)$ 도 입력한다.
3. 단순화 된 도로에서 노드가 속해 있는 회전 시작점과 끝점을 찾아 $dist(N_i)$ 와 $dist(N_{total})$ 를 이용하여 계산한다.

3.9 심벌화(Symbolization)

심벌화 연산자는 주요한 객체들을 심벌화하는 방법이다. 주유소, 아파트, 병원 같이 자주 사용되고, 중요한 객체들의 경우 심벌을 만들어서 해당 데이터를 심벌로 대체함으로써 사용자의 인지 능력을 향상시키고 화면을 단순화하는 기법이다.

심벌은 <그림 2>의 (a)와 같은 도로 특성에 따른 정보와 (b)와 같은 회전에 대한 정보, 건물에 대한 정보 등을 나타낸다.



<그림 2> 주요도로 표현과 회전 방향 표시

4. LMG-tree의 성능 평가

성능 평가는 기존의 알고리즘들인 Reactive-tree, LR-tree와 수행하였다. Reactive-tree는 모든 계층이 하나의 트리로 구성된 경우이고, LR-tree는 각 계층별로 트리가 따로 구성된 경우이다. 두 알고리즘과의 일반화 연산자 중 선택 연산자만을 지원하고 있다. 이에 본 논문에서는 Reactive-tree, LR-tree와의 선택 연산 부분에 대해서만 성능 비교를 수행한다.

둘째, LMG-tree에서의 일반화 연산자 지원 방법이다. LMG-tree가 기존 알고리즘에서 지원하지 않는 나머지 연산자들에 대해서 지원함을 보이기 위해 일반화 연산자에 대한 지원 내용을 기술하였다.

4.1 실험 데이터와 실험 질의문

본 논문에서 사용된 데이터는 서울시 강남구 지역의 45,560건의 실제 데이터를 대상으로 하였다. <그림 3>은 사용된 실제 데이터를 표현한 것이다.



<그림 3> 실험 데이터

실험 질의문은 임의로 1,000개의 영역을 생성하였으며, 각 색인 구조에 같은 조건으로 적용하였다. 그리고, 선택 연산자 수행시에 총 5단계로 계층을 구성하였으며, 각 객체에 대한 계층 값은 임의로 선정하였다.

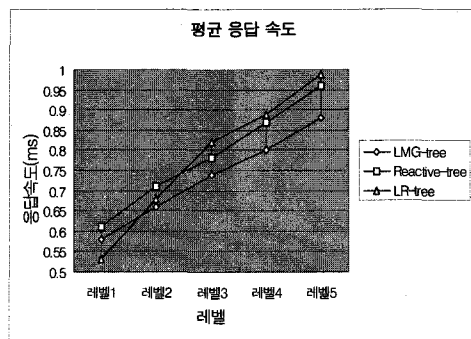
4.2 선택 연산자에 대한 성능 비교

제안한 LMG-tree의 선택 연산자에 대한 성능 평가를 위하여 Reactive-tree, LR-tree와의 성능을 비교하였다. Reactive-tree는 제안한 색인 구조와 같이 계층화된 데이터를 하나의 색인 구조에 저장하는 방식이고, LR-tree는 계층별로 따로 저장하는 방식을 사용하고 있다. 이 두 종류의 색인 방식과 비교함으로써 제안한 색인 구조의 성능을 비교하였다.

성능 평가의 기준은 색인된 데이터의 검색 속도와 메모리 효율성으로 이루어진다. 검색 속도는 평균 응답 속도를 이용하여 비교하며, 메모리 효율성은 전체 노드의 수를 비교함으로써 확인할 수 있다.

4.2.1 평균 응답 속도

평균 응답 속도는 전체 영역 질의에 걸린 시간을 의미한다. 1,000번의 영역 질의에 대해서 인덱스를 탐색하고 객체를 검색하여 화면에 출력할 때까지 걸린 평균 시간을 나타내며, 각 계층별 탐색은 계층 1에서 계층 5까지 순서적으로 진행하였다. 그림 4는 평균 응답 속도에 대한 측정 결과를 나타낸 것이다.



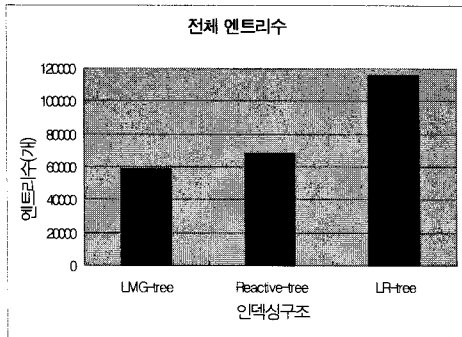
<그림 4> 영역 질의에 대한 평균 응답 속도

계층 1에서는 LR-tree가 가장 좋은 성능을 나타내지만 단계가 깊어 갈수록 LMG-tree의 응답 속도가 더 빨라진다. 그 이유는 저장 방식의 차이로 추정된다. LMG-tree는 영역을 우선으로 하여 계층별로 저장하

지만, LR-tree는 계층을 우선으로 하여 영역별로 저장한다. 따라서, 영역이 검색된 상태에서 계층을 증가시키면 LMG-tree는 하위의 계층만 검색하면 되지만, LR-tree는 해당 계층으로 이동하여 영역 검색을 다시 수행하여야 하기 때문이다. Reactive-tree는 2장에서 기술한 것처럼 중간노드에서 객체가 존재하지 않을 경우 하위 계층의 객체를 재검색함으로써 LMG-tree보다 응답 속도가 느리다.

4.2.2 엔트리수 비교

메모리 용량에 대한 성능을 평가하기 위해서 본 논문에서는 엔트리수를 이용하였다. 객체가 인덱스에 삽입될 때마다 삽입된 엔트리수를 측정하였다. 이렇게 측정된 결과를 <그림 5>에 나타내었다.



<그림 5> 전체 엔트리수 비교

인덱스를 구성하는데 필요한 전체 엔트리의 수는 LMG-tree가 가장 적게 나타났다. 그 이유는 LR-tree의 경우 모든 계층을 따로 저장하기 때문에 가장 많은 엔트리가 사용되고, Reactive-tree의 경우 LMG-tree와 유사하지만 2장에서 기술한 중간 노드에서 비어있는 엔트리가 존재하기 때문이다.

평균 응답 속도와 엔트리수에 대한 실험을 통해서 LMG-tree는 지도 일반화를 지원하는 기존의 공간 색인 기법들에 비해서 검색 성능과 메모리 저장 공간의 효율성 측면에서 모두 우월한 성능을 나타내었다.

5. 결론

핸드폰에서 효율적으로 지도를 표현하기 위해서는 사용자가 쉽게 인식할 수 있도록 지도 데이터의 객체 수를 줄이거나 객체의 모양을 단순화하는 작업이 필

요하다. 그러나, 기존의 공간 색인 기법을 사용할 경우 실시간으로 지도 일반화를 수행하기가 어려우며, 메모리 저장 공간의 비효율성이 나타났다. 또한, 지도 일반화 연산자 중 일부만 지원함으로써 사용자가 인지하기 쉬운 표현이 불가능하였다.

본 논문에서 제안한 LMG-tree는 계층화된 데이터를 저장할 수 있도록 설계하여 불필요한 저장 공간을 제거하였으며, 지도 일반화 연산자들을 지원함으로써 지도 표현을 위한 간략화를 수행할 수 있도록 하였다. 제안한 LMG-tree의 인덱스 구성 방법과 검색, 삽입, 삭제 기법들을 설명하였고 이를 실험을 통해서 성능의 우수성을 증명하였다. 그리고, 지도 일반화를 지원하도록 일반화 연산자들을 설계하였고, 그 예를 나타내었다.

향후 각각의 일반화 연산자들에 대한 추가적인 연구를 통해서 알고리즘에 대한 성능 개선이 필요하며, 사용자가 보다 쉽게 인지할 수 있는 정보 가시화와 관련된 부분에 대한 연구가 필요하다.

참고문헌

- [1] Clodoveu A. Davis Jr., Alberto H. F. Laender, "Multiple Representation in GIS : Materialization Through Map Generalization, Geometric, and Spatial Analysis Operations", ACM-GIS '99, Proceedings of the 7th International Symposium on Advances in Geographic Information Systems, 1999, pp. 60-65.
- [2] Edward P.F. Chan, Kevin K.W. Chow, "On Multi-Scale Display of Geometric Objects", International Journal on Data and Knowledge Engineering 40(1), 2001. pp.91-119.
- [3] P.V. Oosterom, "The Reactive-tree : A Storage Structure for a Seamless, Scaleless Geographic Database", Proceedings of Auto-Carto, 10, 1991, pp.393-407.
- [4] Stefano Spaccapietra, Christine Parent, Christelle Vangenot, "GIS Databases : From Multiscale to MultiRepresentation", Proceedings 4th International Symposium, SARA-2000, Horseshoe Bay, Texas, USA, July 26-29, 2000.
- [5] S. Timpf, "Cartographic Objects in a Multi-Scale

Data Structure", Geographic Information Research : Bridging the Atlantic. 1(1), London, Taylor and Francis, 1997, pp.224-234.

- [6] B. Becker, P. Widmayer, "Spatial Priority Search: An Access Technique for Scaleless Maps", Proceedings of ACM SIGMOD, Denver, Colorado, 1991, pp.128-137.
- [7] 권준희, 윤용익, "LR 트리: 지도 일반화를 지원하는 공간 데이터를 위한 공간 색인", 정보처리학회 논문지D 제9권-D권 제4호, 2002, pp.543-554.
- [8] W. Peng, K. Tempfli, M. Molenaar, "Automated Generalization in a GIS Context", GeoInformatics 96, Vol. 1, 1996, pp.135-144.
- [9] W. Yang, C. Gold, "A System Approach to Automated Map Generalization, Proceedings, International Workshop on Dynamic and Multi-Dimensional GIS, pp.229-235
- [10] A. Guttman, "R-trees : A Dynamic Index Structure for Spatial Searching", Proceedings of the ACM SIGMOD International Conference on Management of Data, Boston, Ma., 1984, pp.47-54.
- [11] T.Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-tree : A Dynamic Index for Multidimensional Objects", Proceedings of the 13th International Conference on VLDB, Brighton, England, 1987, pp.507-518.
- [12] N. Beckmann, H.P. Kriegel, R.Schneider, and B. Seeger, "The R*-tree : an Efficient and Robust Access Method for Points and Rectangles", Proceedings of ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, 1990, pp.322-331.
- [13] Hutflesz, A. Six, H-W, and P. Widmayer, "The R-File: An Efficient Access Structure for Proximity Queries", Proceedings of IEEE 6th International Conference on Data Engineering, 1990, pp.372-379.
- [14] I. Kamel, C. Faloutsos, "Hilbert R-Tree: An Improved R-Tree Using Fractals", Proceedings of 20th VLDB, 1994, pp.500-509.
- [15] D.H. Douglas, T.K. Peucker, "Algorithms for the Reduction of Points Required to Represent a

Digitized Line or its Caricature", Canadian Cartography, 10, 1973, pp.112-122.



이기정

1999년 서울시립대학교 학사(국사학)
2003년 경원대학교 일반대학원 석사
(전자계산학)

2004년 ~ 현재 경원대학교 일반대학원
전자계산학과 박사과정

관심분야 : 컴퓨터그래픽스, 모바일 GIS



황보택근

1983년 고려대학교 공과대학 공학사
1987년 CUNY Computer Science 석사
1995년 S.I.T. Computer Science 박사
1988년 ~ 1993년 Q-Systems,
Senior Technical Staff

1995년 ~ 1997년 삼성종합기술원 선임연구원
1997년 ~ 현재 경원대학교 소프트웨어대학
소프트웨어학부 부교수

관심분야 : 컴퓨터그래픽스, 영상처리, GIS 등



양영규

1972년 서울대학교 학사
1974년 서울대학교 석사
1985년 Texas A&M Univ. Ph.D
(Remote Sensing)

1973년 ~ 1996년 시스템공학연구소 책임연구원
1996년 ~ 2003년 한국전자통신연구원 책임연구원
2002년 ~ 2003년 University of California at Irvine
객원연구원

2003년 ~ 현재 경원대학교 소프트웨어대학 교수,
전산정보원장

관심분야: LBS, GIS/RS, 텔레매틱스