

GR-tree: 무선 센서 네트워크에서 에너지 효율적인 분산 공간색인기법

The GR-tree: An Energy-Efficient Distributed Spatial Indexing Scheme in Wireless Sensor Networks

김민수* 장인성**
Min Soo Kim In Sung Jang

요약 최근 특정 공간영역 내에 포함되는 센서노드들만의 센싱정보를 에너지 효율적으로 수집하는 센서 네트워크 기반 공간질의처리에 대한 관심이 증가하고 있다. 이러한 센서 네트워크 기반 공간질의처리의 가장 단순한 방법은 모든 센서노드의 위치와 센싱정보를 서버로 수집한 다음, 서버에서 공간질의를 처리하는 Centralized 방법이다. 이 방법은 간단하다는 장점은 있지만, 모든 센서노드를 접근하기 위하여 소요되는 높은 무선통신 비용으로 인하여 센서노드의 에너지 효율성이 크게 떨어지는 단점을 가지고 있다. 이러한 Centralized 방법을 보완하기 위하여 센서노드에서 분산 공간 필터링을 수행하여 센서노드 간의 무선통신 횟수를 감소시키는 In-network 기반 분산 공간색인기법들이 제안되어 왔다. 그러나, 이러한 분산 공간색인기법들은 대부분 서버에서 이용되던 기존 공간색인기법들을 센서 네트워크에 단순히 적용하였기 때문에, In-network 환경에서 공간 필터링의 효과와 센서노드들 간의 무선 라우팅을 동시에 최적화하지 못하는 단점을 가지고 있다. 이에 본 논문에서는 In-network 환경에서 공간 필터링을 최적화 하면서 동시에 센서노드들 간의 라우팅을 보장할 수 있는 GR-tree의 새로운 분산 공간색인기법을 제안하고자 한다. GR-tree 방법은 R-tree와 유사하게 MBR 기반의 트리를 구성하며, 센서노드들 간의 무선 라우팅 및 공간적인 인접성을 보장하면서 MBR들 간의 겹침을 최소화할 수 있는 특징을 가지고 있다. 끝으로, GR-tree와 기존 방법들의 다양한 성능 비교 실험을 통하여 제안된 방법의 효율성을 보여주고자 한다.

키워드 : 센서 네트워크 데이터베이스, In-network 공간 질의, 분산 공간색인, GR-tree

Abstract Recently, there has been much interest in the spatial query which energy-efficiently acquires sensor readings from sensor nodes inside specified geographical area of interests. The centralized approach which performs the spatial query at a server after acquiring all sensor readings, though simple, it incurs high wireless transmission cost in accessing all sensor nodes. In order to remove the high wireless transmission cost, various in-network spatial indexing schemes have been proposed. They have focused on reducing the transmission cost by performing distributed spatial filtering on sensor nodes. However, these in-network spatial indexing schemes have a problem which cannot optimize both the spatial filtering and the wireless routing among sensor nodes, because these schemes have been developed by simply applying the existing spatial indexing schemes into the in-network environment. Therefore, we propose a new distributed spatial indexing scheme of the GR-tree. The GR-tree which forms a MBR-based tree structure, can reduce the wireless transmission cost by optimizing both the efficient spatial filtering and the wireless routing. Finally, we compare the existing spatial indexing scheme through extensive experiments and clarify our approach's distinguished features.

Keywords : Sensor Network DB, In-network Spatial Query, Distributed Spatial Index

† 본 연구는 국토해양부 첨단도시개발사업 - 지능형국토정보기술혁신사업과제의 연구비지원(07국토정보C05)에 의해 수행되었음.

* 한국전자통신연구원 공간정보연구팀 책임연구원 minsoo@etri.re.kr

** 한국전자통신연구원 공간정보연구팀 선임연구원 e4dol2@etri.re.kr(교신저자)

1. 서론

무선 센서 네트워크 환경에서 각 센서노드는 제한된 배터리 용량, 제한된 통신 대역폭, 그리고 제한된 컴퓨팅 능력을 가지고 있다. 이 중에서도 특히 제한된 배터리 용량은 센서노드의 동작과 직접적인 연관성을 가지고 있기 때문에, 지금까지 센서노드의 에너지 소모를 최소화시키기 위한 다양한 연구가 진행되어 왔다[18]. 예를 들어, 데이터베이스 분야에서는 센서 네트워크를 가상의 데이터베이스 테이블로 간주하고 그 테이블로부터 센싱정보를 획득할 때 센서노드들의 에너지 소모를 최소화하기 위한 Cougar [1], TinyDB[9]와 같은 기술들이 제안되어 왔다.

최근에는 모든 센서노드로부터 센싱정보를 획득하는 단순 질의처리 이외에, 특정 공간영역 내에 포함되는 센서노드들의 센싱정보를 획득하는 센서 네트워크 기반 공간질의처리에 대한 관심이 높아지고 있다. 이러한 공간질의를 처리하기 위한 가장 단순한 방법으로 서버기반의 Centralized 질의처리 방법이 있다. Centralized 방법은 모든 센서노드들의 위치와 센싱정보를 서버로 수집한 다음에, 서버에서 공간질의를 처리하는 방법이다. 이 방법은 센서 네트워크에 적용하기가 매우 용이하다는 장점을 가지고 있지만, 모든 센서노드들을 접근하는데 있어서 센서노드의 에너지 효율성이 크게 떨어지는 단점을 가지고 있다. 이러한 Centralized 방법을 보완하기 위한 방법으로 In-network 질의처리 방법들이 제안되었는데, 이 방법은 공간질의가 서버가 아니라 센서노드에서 분산되어 처리되는 특징을 가지고 있다. 구체적으로 In-network 질의처리 방법과 관련하여 지금까지 다양한 분산 공간질의처리 알고리즘 [6, 7, 8, 11, 12, 15, 16, 17] 및 분산 공간색인기법들 [3, 4, 14]이 제안되어 왔다. In-network 방법은 센서노드에서 공간 필터링을 수행하여 센서노드 간의 무선통신 횟수를 감소시킴으로써, Centralized 방법에 비하여 센서노드의 에너지 소모를 감소시키는 효과를 보여주었다. 그러나, 지금까지의 In-network 방법은 대부분 서버 기반 공간 질의처리기에서 이용되던 알고리즘을 센서 네트워크에 단순 적용하였기 때문에 In-network 환경에서 공간 필터링의 효과는 얻을 수 있지만, 센서노드들 간의 라우팅을 동시에 최적화하지 못하는 단점을 가지고 있었다[3, 4, 10].

이에 본 논문에서는 센서 네트워크 환경에서 공간

질의를 처리하고자 할 때, 공간 필터링을 최적화하여 센서노드들의 에너지 소모를 최소화하면서 동시에 센서노드들 간의 라우팅을 보장할 수 있는 새로운 분산 공간색인기법인 GR-tree를 제안하고자 한다. 제안된 분산공간색인 방법은 R-tree 방법과 같이 MBR 기반의 트리를 구성하며, 센서노드들 간의 무선 라우팅 및 공간적인 인접성을 보장하면서 MBR들 간의 겹침을 최소화하기 위하여 Grid 기반의 클러스터링 방법을 동시에 이용하고 있다.

본 논문은 다음과 같이 구성되어 있다. 2장의 관련 연구에서는 센서 네트워크 기반의 기존 공간질의 처리 방법들에 대하여 살펴보고, 3장에서는 본 논문에서 제안하는 GR-tree (Grid-based R-tree)의 분산 공간색인기법에 대하여 자세히 살펴볼 것이다. 4장에서는 다양한 유형의 공간질의처리에 대하여 Centralized 방법, 기존의 분산 공간색인기법 그리고, 새로이 제안된 GR-tree 방법의 공간검색 성능을 비교분석 할 것이다. 끝으로 5장에서는 본 논문의 결론 및 연구 방향에 대하여 살펴볼 것이다.

2. 관련 연구

초기에 센서 네트워크 환경에서 공간질의를 수행하기 위해서는 In-network 환경에서 일반적인 센서 네트워크 라우팅을 통하여 모든 센서노드의 센싱정보를 서버로 수집한 다음에, 서버에서 R-tree[5], R*-tree[2]와 같은 기존의 공간색인기법을 적용하여 공간질의를 수행하는 Centralized 방법을 주로 이용하였다. 그러나, 이러한 Centralized 방법은 In-network 환경에서 공간 라우팅 기반의 공간 필터링을 전혀 수행하지 않기 때문에 공간질의를 수행할 때 모든 센서노드를 접근해야 하고, 이로 인하여 발생하는 무선통신 비용은 전체 센서 네트워크의 에너지 소모를 매우 크게 만드는 문제점을 가지고 있었다.

그리하여, 센서 네트워크에서 공간질의를 수행할 때 센서노드의 에너지 소모를 최소화하기 위한 다양한 In-network 기반의 공간색인기법들이 제안되었는데, Demirbas et al.[3]은 공간 데이터베이스의 대표적인 R-tree[5]의 공간색인을 그대로 In-network의 공간색인기법으로 적용한 Peer-tree의 방법을 제안하였다. 이 방법은 In-network 환경의 센서노드들에 대하여 분산 R-tree를 구성하기 위하여 부모와 자식에 해당되는 센서노드들이 거리에 관계없이 무

조건 직접적으로 연결이 가능하다는 가정을 하고 있다. 그러나, 근거리 무선통신으로 구성된 센서 네트워크 환경에서 센서노드들은 직접적으로 연결되기 보다는 멀티 홉 통신을 통하여 간접적으로 연결된다. 그러므로, 이 방법은 이론적으로는 가능한 방법이나, 실제 센서 네트워크 환경에 적용하기에는 제약사항이 많은 방법이라고 할 수 있다.

Soheili et al.[14]은 R-tree의 기본 개념을 이용하여 In-network 환경에서 효율적으로 동작 가능한 SPIX의 방법을 제안하였다. SPIX 방법은 Peer-tree와 달리 분산된 센서노드들에 대하여 공간적인 라우팅을 형성하고, 이러한 공간적인 라우팅을 기반으로 R-tree의 MBR(Minimum Bounding Rectangle)을 이용하여 계층적인 공간색인 트리를 형성하게 된다. SPIX의 실험결과는 이 방법이 센서노드들에 대한 공간검색에서 효율적임을 보여주고 있다. 그러나, 이 방법은 R-tree의 MBR 기반 클러스터링 방법을 공간적인 라우팅을 동시에 고려하면서 적용할 때, 색인트리에서 인접한 센서노드들의 MBR들 간의 Overlap이 크게 증가할 위험성을 가지고 있다. 그리고, 이러한 MBR들 간의 Overlap 증가는 공간검색의 성능을 저하시키는 단점을 유발시킨다.

Demirbas et al.[4]와 Meka et al.[10]은 기존의 Quad-tree의 공간색인기법을 In-network에 분산하여 적용한 DQT (Distributed Quad-Tree)와 DIST (Distributed Spatio-Temporal Index) 방법을 제안하였다. 이들 방법들의 실험결과를 보면 공간질의에서 효율적인 검색 성능을 보여줄 수 있다. 그러나, 이 방법들은 각 Quadrant에 리더 역할을 하는 센서노드들이 존재하고, 이들 리더 역할을 하는 센서노드들은 다음 레벨의 리더 센서노드들이 멀리 떨어져 있다할지라도 무선 라우팅이 항상 가능함을 가정하고 있다. 그러므로 이 방법들도 비록 효율적인 공간검색 성능을 보여준다 할지라도 Peer-tree 방법과 마찬가지로 실제 센서 네트워크 환경에 적용하기에는 어려운 제약사항을 가지고 있다.

이에 본 논문에서는 센서노드들 간의 실질적인 무선 라우팅을 고려하면서도 In-network 환경에서 센서노드에 분산되어 에너지 효율적으로 동작할 수 있는 새로운 분산 공간색인기법을 제안하고자 한다.

3. In-network 기반 분산 공간색인기법

3.1 In-network 기반 분산 공간색인의 특징

공간 데이터베이스의 대표적인 R-tree 계열의 공간색인기법을 그대로 센서 네트워크의 공간색인기법으로 적용하기에는 다음과 같은 문제가 있다.

첫째, R-tree 계열의 공간색인기법은 객체들의 공간(위치)정보를 이용하여 인접한 객체들을 MBR을 기반으로 클러스터링하고, 클러스터링된 MBR들에 대하여 계층적으로 클러스터링을 수행하여 MBR 기반의 트리를 형성한다. 여기서, MBR 기반의 클러스터링은 클러스터링을 수행하는 시스템이 모든 객체들의 공간정보를 미리 알고 있어야 한다. 그러나, 센서 네트워크에서의 공간색인기법은 색인정보가 각 센서노드에 분산되어 저장되고, 각 센서노드는 자신의 인접 센서노드의 공간정보만을 알 수 있기 때문에 기존 R-tree 계열의 MBR 기반 클러스터링 방법을 그대로 적용하는 것이 불가능하다.

둘째, R-tree 계열의 클러스터링 방법은 기본적으로 객체들 사이의 공간적인 인접성을 고려하여 수행되는 반면에, 센서 네트워크에서의 클러스터링 방법은 센서노드들의 공간적인 인접성뿐만 아니라 센서노드들 간의 연결 그리고 센서노드와 베이스스테이션까지의 연결을 보장하기 위한 무선 라우팅도 함께 고려해야 한다.

이러한 센서 네트워크 기반 분산 공간색인의 특징을 반영하여 SPIX[14]의 분산 공간색인기법이 제안되었다. SPIX 방법은 그림 1에서 보듯이 단말 센서노드로부터 베이스스테이션까지 무선연결을 고려하면서 MBR 기반 클러스터링을 계층적으로 적용하여 R-tree와 유사한 트리 구조를 형성하게 되며, 트리를 구성하는 색인정보는 각 센서노드에 분산하여 저장된다. 예를 들어, 센서노드 3은 센서노드 1과 2를 자식 노드로 유지하고 MBR A의 정보를 저장하게 되며, 센서노드 5는 센서노드 3과 4를 자식 노드로 유지하고 MBR B의 정보를 저장하게 된다.

SPIX 방법은 MBR 기반 클러스터링과 무선 라우팅을 동시에 고려함으로써 In-network 환경에서 R-tree와 유사한 분산 공간색인을 지원할 수 있다. 그러나, 이 방법은 그림 1에서 보듯이 공간색인 트리가 단말 센서노드로부터 베이스스테이션으로 가까워질수록 MBR의 크기가 점차적으로 커지기 때문에 공간색인 트리에서 이웃하는 MBR들 간의

Overlap이 점차적으로 커지게 되는 위험을 가지고 있다. MBR들 간의 Overlap 증가는 In-network에서 공간검색의 성능을 급격히 떨어뜨리게 된다.

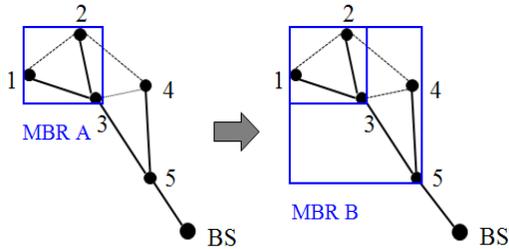


그림 1. SPIX의 공간색인 구성방법

3.2 GR-tree의 분산 공간색인기법

이에 본 논문에서는 MBR 기반 트리를 이용하여 분산 공간색인을 구성하고자 할 때, 센서노드들 간의 무선 라우팅을 보장하면서 베이스스테이션 근처 MBR들 간의 Overlap을 최소화할 수 있는 GR-tree (Grid-Based Rectangle tree)의 새로운 공간색인기법을 제안하고자 한다. GR-tree 방법은 기본적으로 SPIX와 마찬가지로 공간색인정보가 센서노드에 분산 저장되며, 저장되는 색인정보의 구성은 아래와 같이 R-tree와 유사하다.

$(ID, MBR, Grid\ ID, Depth, Entry)$

여기서, ID 는 센서노드의 Identifier 정보를, MBR 은 센서노드 자신과 자식 센서노드들을 포함하는 MBR 정보를, $Grid\ ID$ 는 센서노드가 위치하는 Fixed Grid 상에서의 주소 정보를, $Depth$ 는 GR-Tree에서 센서노드 자신의 Level 정보를 의미하며, 끝으로 $Entry$ 는 자식 센서노드들에 대한 정보를 의미한다.

$Entry = (E_1, E_2, \dots, E_n)$

여기서, n 은 자식 센서노드들의 수를 의미하며, E_i 는 자식 센서노드 i 의 ID 와 MBR 정보로 구성된다.

In-network 기반 GR-tree의 생성방법은 앞에서 설명한 바와 같이 R-tree 계열의 색인 생성방법과는 크게 다르며, SPIX와 유사하게 Advertisement와 Parent Selection의 두 과정을 통하여 생성된다.

Advertisement 과정은 각 센서노드에 대하여 부모 또는 자식 센서노드가 될 수 있는 후보 센서노드 리스트를 결정하는 과정으로 베이스스테이션으로부터 시작된다. 베이스스테이션은 후보가 될 수 있는

자식 센서노드를 찾기 위하여 자신의 Advertisement 메시지를 통신 가능한 환경 내에 있는 모든 센서노드들에게 브로드캐스팅하고, 베이스스테이션으로부터 메시지를 수신한 센서노드는 자신의 Advertisement 메시지를 하위의 센서노드들에게 반복적으로 브로드캐스팅 한다. Advertisement 과정은 모든 센서노드가 메시지를 받을 때까지 수행되며, 이 과정이 종료된 후에는 베이스스테이션을 포함한 모든 센서노드들이 자신의 부모 및 자식 센서노드가 될 수 있는 후보 센서 리스트를 결정하게 된다. 여기서, 임의의 센서노드 i 의 Advertisement 메시지 $AD(i)$ 는 다음과 같이 구성되며,

$AD(i) = (i, Depth, L_i, L_B)$

L_i 와 L_B 는 각각 센서노드 i 와 베이스스테이션의 기하학적인 위치정보를 의미한다. Algorithm 1은 임의의 센서노드 i 가 j 로 $AD(i)$ 를 전달하였을 때 수행되는 Advertisement 과정과 j 가 주변의 연결 가능한 센서노드 리스트 $SL(j)$ 에 대하여 자신의 $AD(j)$ 를 반복적으로 전달하는 과정을 보여준다.

Algorithm 1: Advertisement ($AD(i)$)

```

1: If ( $AD(i)$  exists in the candidate child list of  $j$ ) Then
2:   Return;
3: Else
4:    $CPL(j) \leftarrow j$  adds  $AD(i)$  into its candidate parent list
5:    $CCL(i) \leftarrow i$  adds  $AD(j)$  into its candidate child list
6:    $GA(j) \leftarrow$  Calculate a grid address of  $j$ 
7:    $TD(j) \leftarrow$  Set a tree depth of  $j$  to  $TD(i)+1$ 
8:   For each sensor node  $S$  of  $SL(j)$  Do
9:      $S.Advertisement(AD(j))$ 
10:  End For
11: End If

```

Algorithm 1은 라인 1에서 보듯이 두 센서노드 i 와 j 사이에 Cross Advertisement를 허용하지 않고 있는데, 이는 GR-tree에서 사이클의 발생을 방지하기 위함이다. 라인 4-5에서는 노드 j 와 i 가 각각 자신의 후보 부모와 후보 자식 리스트로서 i 와 j 를 추가하고 있다. 그리고, 라인 6에서 센서노드 j 의 Grid 주소는 전체영역의 크기 정보와 GR-tree에서 이용되는 Fixed Grid의 크기 정보를 이용하여 쉽게 계산될 수 있다. 구체적으로 본 연구에서는 센서노드의 통신거리를 이용하여 Fixed Grid의 크기를 결정하고 있으며, 센서노드가 설치되는 전체 영역의 위치정보

는 미리 알고 있으며, 각 센서노드는 자신의 위치정보를 측정할 수 있다고 가정하고 있다. 이러한 Grid 정보는 추후 GR-tree의 Parent Selection 과정에서 이용된다.

그림 2는 임의의 센서노드 집합 $\{0, 1, \dots, 7\}$ 에 대하여 Algorithm 1의 수행 후에 후보 부모노드 리스트가 어떻게 생성되는지를 보여준다. 그림 2를 보면 BS에서 시작된 Advertisement 과정이 일차적으로 센서노드 $\{1, 2, 3\}$ 으로 전해지고, 이후 센서노드 1은 $\{4, 5\}$ 로 그리고 센서노드 3은 $\{6, 7\}$ 순으로 계속하여 Advertisement가 브로드캐스팅 됨으로써 각 센서노드는 다음과 같은 후보 부모노드 리스트를 얻게 됨을 알 수 있다.

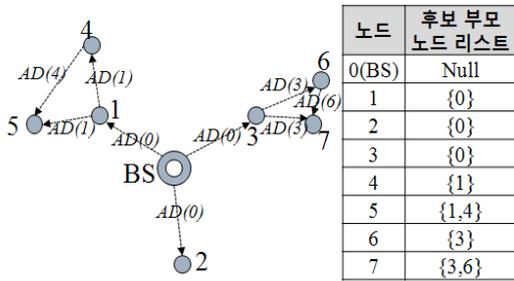


그림 2. 임의의 센서노드 집합 $\{0, 1, \dots, 7\}$ 에 대하여 Advertisement 알고리즘의 수행 결과

Parent Selection 과정은 각 센서노드가 후보 부모 센서 리스트에 대하여 자신과의 공간적인 인접성과 무선 라우팅을 동시에 고려하면서 자신의 부모를 선택하는 과정을 의미한다. 이러한 Parent Selection 과정은 Advertisement 과정과는 반대로 단말 센서노드로부터 시작되며, 모든 센서노드가 자신의 부모를 선택하고 최종적으로 베이스스테이션이 부모노드로 선택될 때 종료된다. Algorithm 2는 임의의 단말 센서노드 j 에서 수행되는 Parent Selection의 구체적인 과정을 보여준다.

Algorithm 2의 라인 1-13은 센서노드 j 에서 자신과 동일한 Grid 내에 포함되어 있는 후보 부모 센서 리스트 $CPL(j)$ 로부터 부모를 선택하는 과정을 수행하는데, 이는 GR-tree 색인 구조에서 가능한 센서노드들 사이의 공간적인 인접성을 최대한 보장하기 위함이다. 만약, 센서노드 j 와 동일한 $GA(j)$ 내에 후보 센서가 존재하지 않은 경우에는 라인 14-28과 같이 $GA(j)$ 의 인접한 Grid들에 포함되어 있는 후보

Algorithm 2: ParentSelection ()

```

1: For each sensor node CP of CPL(j) Do
2:   If ( GA(CP) == GA(j) ) Then
3:     If ( TD(CP) < TD(j) ) Then
4:       P ← CP is selected as the parent node of j
5:     Else If ( TD(CP) == TD(j) ) Then
6:       DBS ← Calculate distance between j and BS
7:       DCPBS ← Calculate distance between CP and BS
8:       If ( DCPBS <= DBS ) Then
9:         P ← CP is selected as the parent node of j
10:      End If
11:    End If
12:  End If
13: End For
14: If ( P is null ) Then
15:   For each sensor node CP of CPL(j) Do
16:     If ( GA(CP) is one of adjacent grids of GA(j) ) Then
17:       If ( TD(CP) < TD(j) ) Then
18:         P ← CP is selected as the parent node of j
19:       Else If ( TD(CP) == TD(j) ) Then
20:         DBS ← Calculate distance between j and BS
21:         DCPBS ← Calculate distance between CP and BS
22:         If ( DCPBS <= DBS ) Then
23:           P ← CP is selected as the parent node of j
24:         End If
25:       End If
26:     End If
27:   End For
28: End If
29: CL(P) ← P adds j into its child list
30: P expands its MBR so as to include the MBR of j
    
```

부모 센서 리스트로부터 부모를 선택하는 과정을 수행되게 된다. 여기서, 최대 8개로 구성되는 인접 Grid들에 대한 Parent Selection 과정은 GR-tree 색인을 구성하는 MBR들 간의 Overlap을 최소화하기 위하여 특별히 지정된 Grid 순서에 따라서 수행된다. 최적의 경우, 그림 3과 같이 GR-tree의 이웃 MBR들 간의 Overlap이 완전히 제거된 GR-tree 색인이 생성될 수 있는데, 그림 3을 보면 BS는 서로 겹침이 전혀 존재하지 않는 A, B, C의 MBR을 가지는 센서노드를 자식 노드로 유지하고 있다. 이와 같

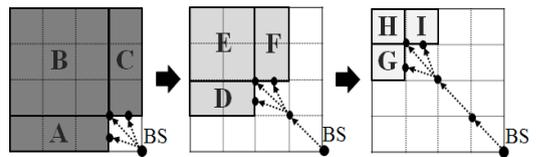


그림 3. 공간색인 트리에서 이웃 MBR들 간의 겹침이 없는 최적화된 GR-tree 구성 예제

이 A, B, C의 MBR을 형성하기 위해서는 계층적으로 A와 C에 속한 센서노드들은 BS에 대한 직선 방향을, B에 속한 센서노드들은 BS에 대한 대각선 방향을 우선순위로 하여 부모 노드를 선택하고 있다. 하위계층인 MBR B를 가지는 센서노드도 마찬가지로 서로 겹침이 존재하지 않는 D, E, F의 MBR을 자식 노드로 유지하고 있으며, MBR E를 가지는 센서노드도 서로 겹침이 존재하지 않는 G, H, I의 MBR을 자식 노드로 유지하고 있음을 알 수 있다. 이와 같이 이웃하는 MBR들 간의 겹침을 최소화하는 GR-tree를 구성하기 위해서 본 연구에서는 Algorithm 2의 라인 14-28을 수행할 때, 그림 4와 같이 특별히 정해진 순서에 의하여 최대 8개의 인접 Grid를 검색하도록 하고 있다. 구체적으로 그림 4를 보면, Grid A와 같이 대각선상의 Grid에 속한 센서들은 대각선 방향을 최우선 순위로 하여 그림 4(a)와 같이 인접 Grid에 대하여 부모 센서를 검색하면 된다. Grid B와 같이 대각선 위의 Grid에 속한 센서들은 y축 아래 방향을 최우선 순위로 하여 그림 4(b)와 같이 인접 Grid를 검색하면 되며, Grid C와 같이 대각선 아래의 Grid에 속한 센서들은 x축 오른쪽 방향을 최우선 순위로 하여 그림 4(c)와 같이 인접 Grid를 검색하면 된다.

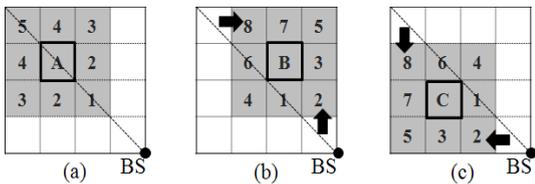


그림 4. Parent Selection에서 인접 Grid 검색 순서

끝으로, 라인 29-30에서는 부모노드 P 에 j 를 자식으로 등록시키고, P 의 색인정보인 MBR을 확장하고 있다.

그림 2의 Advertisement 결과에 대하여 Parent Selection 알고리즘을 수행한 후의 GR-tree는 그림 5와 같다. 그림 5를 보면 BS는 센서노드 {1, 2, 3}을 자식노드로 가지고 있으며, 센서노드 1과 3은 각각 {4, 5}와 {6, 7}을 자식노드로 가지고 있음을 알 수 있다. 이는 Parent Selection 알고리즘을 통하여 그림 2에서의 센서노드 5가 자신의 부모로 센서 4 보다는 1을 선택하였으며, 센서노드 7은 센서 6 보다는 3을 선택하였음을 의미한다.

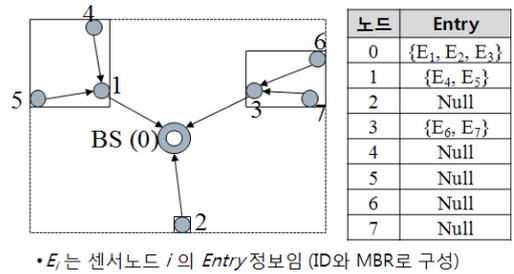


그림 5. 임의의 센서노드 집합 {0, 1,...,7}에 대하여 Parent Selection 알고리즘 수행 결과

Algorithm 2에서 설명한 바와 같이 제안된 GR-tree 기법은 인접한 MBR들 간의 겹침을 최소화 하면서 가능하면 공간색인 트리의 Depth를 최소화 하는 전략을 동시에 이용하고 있는데, 이러한 인접 MBR 간 겹침 최소화 및 트리 Depth 최소화 전략은 센서노드에 대한 공간검색 과정에서 기존 기법에 비하여 좋은 성능을 보여준다.

본 연구에서는 기본적으로 이미 고정적으로 설치된 센서노드들에 대하여 GR-tree를 생성하는 것을 가정하고 있다. 그러나, 새로운 센서노드가 추가되거나 기존 노드가 고장나는 경우를 대비하여 GR-tree는 센서노드의 Insertion 및 Deletion 과정을 수행해야 하는데, 이러한 과정들은 GR-tree 생성을 위한 기존 Advertisement와 Parent Selection 알고리즘을 이용하여 간단히 수행이 가능하다. 예를 들어, 새로운 센서노드 n 이 추가되었을 때, n 은 자신의 메시지를 통신 범위에 있는 인접한 노드들에게 브로드캐스팅 한다. 이후부터는 GR-tree 생성 알고리즘과 동일하게, n 의 메시지를 수신한 노드들이 Advertisement 알고리즘을 수행하고, 센서노드들로부터 수집한 Advertisement 메시지를 이용하여 n 은 Parent Selection 알고리즘을 수행하여 자신의 부모를 결정하면 된다.

Deletion 알고리즘은 Insertion 알고리즘 보다 조금 더 복잡하게 수행된다. 첫째, 어떠한 센서노드에 Failure가 발생해서 Deletion 되어야 하는지를 찾아야 한다. 이를 위하여, 본 연구에서는 센서 네트워크에서 일반적으로 센서노드 Failure를 발견하기 위하여 부모와 자식 간의 관계를 특정 시간을 두고 주기적으로 조사하는 방법을 이용한다[14]. 둘째, Failure된 센서노드가 검색되면, 그 노드를 GR-tree에서 삭제하면 되는데, Failure된 센서노드는 동작을 하지

않기 때문에 특별한 알고리즘을 수행하지 않아도 GR-tree의 라우팅 과정에서 자동으로 삭제가 된다. 그러나, Failure된 노드의 자식노드들도 GR-tree에서 같이 삭제가 되는 문제가 발생하기 때문에 자식노드들은 이전에 보유하고 있던 자신의 후보 부모리스트를 이용하여 각각 Parent Selection 알고리즘을 다시 수행하여 새로운 부모 노드를 찾는 과정을 수행해야 한다.

끝으로, 이러한 Insertion 및 Deletion 알고리즘은 부모노드의 MBR 정보 갱신을 위한 Propagation 작업으로 인하여 많은 무선통신 비용도 소모되기 때문에 잦은 Insertion 및 Deletion은 GR-tree의 유지보수 비용을 크게 증가시키는 문제점이 있다.

3.3 GR-tree 기반 공간검색

GR-tree 기반 공간검색은 R-tree와 유사한 방식으로 수행된다. R-tree가 트리의 루트로부터 공간검색을 시작하듯이, GR-tree에서는 베이스스테이션으로부터 공간검색이 시작되며, 특정 노드 j 에서 검색 영역 Area에 대한 공간검색 과정은 다음과 같다.

Algorithm 3: Searching (Area)

```

1: If (  $j$  is within Area ) Then
2:   Transfer information of  $j$  to its parent
3: End If
4: If (  $j$  is a leaf sensor node ) Then
5:   Return;
6: Else
7:   For each child  $C$  of  $CL(j)$  Do
8:     If ( Area is overlapped with  $C$  ) Then
9:        $C$ .Searching(Area)
10:    End If
11:  End For
12: End If

```

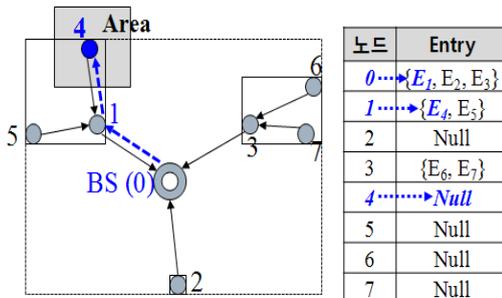


그림 6. 공간검색 영역 Area에 대한 검색 예제

Algorithm 3의 라인 2는 j 에 대한 공간검색 결과를 부모 센서를 통하여 계층적으로 BS까지 전달하도록 구현된다.

그림 6은 공간검색 영역 Area가 주어졌을 때, Algorithm 3에 따라 Area가 BS로부터 자식노드인 1의 Entry인 E_1 , 그리고 1의 자식노드인 4의 Entry인 E_4 와의 겹침이 존재하는지를 계층적으로 조사하여 검색결과로 센서노드 4를 찾는 과정을 보여준다.

4. 성능 평가 및 분석

4.1 실험 환경

성능 평가 및 분석에서는 제안된 GR-tree 방법과 기존의 SPIX 방법, Parent Selection 과정에서 베이스스테이션에 가장 가까운 센서노드를 무조건적으로 부모로 선택하는 Closest Parent Selection(CPS) 방법 그리고 분산 공간색인을 이용하지 않는 Centralized 방법의 성능을 비교 분석 하였다. 여기서, SPIX 방법은 부모 센서를 선택할 때 확장되는 MBR 면적의 최소화를 이용하는 SPIX(Area) 방법과 확장되는 MBR 둘레 길이의 최소화를 이용하는 SPIX(Perimeter) 방법의 두 가지에 대하여 실험이 수행되었다. 공간검색의 성능비교를 위한 기준으로 공간검색 과정에서 소모되는 센서노드 간의 무선통신 횟수를 이용하였는데, 이는 센서노드에서 무선통신이 CPU 프로세싱 또는 센싱 작업에 비하여 가장 많은 에너지를 소모하기 때문이다[13]. 구체적으로 임의의 공간질의 Q를 In-network 환경의 센서노드들에게 전송하기 위하여 소모되는 무선통신 횟수($T_{sending}$)와 공간질의 수행 후의 결과를 수집하기 위하여 소모되는 무선통신 횟수($T_{receiving}$)를 측정하여 성능을 비교 분석하였다. 본 연구에서는 정확한 성능실험을 위하여 GR-tree, SPIX, CPS와 Centralized의 알고리즘들을 Java를 이용하여 구현하였으며, 이외에 별도로 센서 네트워크 기반의 다양한 공간질의 수행 및 무선통신 횟수를 측정할 수 있는 시뮬레이션 프로그램도 Java를 이용하여 구현하였다. 성능 평가는 Intel Core i7 CPU 2.67GHz, 3GB RAM의 Microsoft Windows XP의 시스템 환경에서 Eclipse Europa와 JDK 1.6을 이용하여 수행되었다. 끝으로, 시뮬레이션 환경에서 센서노드가 설치되는 전체 영역은 2,000×2,000 meters이고, 공간검색 및 결과 수집 과정에서 센서노드 간 통신에 사용되는 무선통신

패킷의 용량은 MBR 형태의 공간검색 영역 정보 및 취합된 센싱결과 정보를 저장하기에는 충분한 크기이고, 모든 센서노드의 통신 가능한 거리는 100 meters로 동일하다고 가정하였다.

4.2 실험 결과 및 분석

본 성능 평가 및 분석에서는 다양한 환경에서의 GR-tree 방법의 효율성을 검증하기 위하여 센서노드의 수 및 공간질의 영역을 변경시키면서 실험을 수행하였다.

첫 번째 실험에서는 센서노드의 수가 다양하게 변경될 때, GR-tree 방법과 SPIX(Area), SPIX (Perimeter), CPS 공간색인기법의 성능을 비교 분석하였다. 본 실험에서는 2,000 x 2,000 meters의 전체 영역에 센서노드의 수를 2,000개에서 20,000개로 증가시켰으며, 공간질의 영역의 크기는 500 x 500 meters로 고정하였다. 단, 공간질의 영역의 위치에 따라서 성능이 다르게 나타날 수 있기 때문에 그림 7과 같이 베이스스테이션을 중심으로 공간질의 수행에서 무선통신 횟수가 상이할 가능성이 있는 네 군데의 위치에 대하여 실험을 수행하였다. 베이스스테이션으로부터 가장 먼 질의 영역 (a), 가장 가까운 질의 영역 (c), 중앙의 질의 영역 (d), 그리고 나머지 질의 영역 (b)에 대하여 실험을 수행하였다. 끝으로, 첫 번째 실험에서는 공간색인을 이용하지 않는 Centralized 방법은 실험대상에서 제외하였다. 이는 2,000 x 2,000 meters의 전체 센싱정보를 모두 수집해야 하는 Centralized 방법과 분산 공간색인을 이용하여 500 x 500 meters의 한정된 공간의 센싱정보만을 수집하는 방법과의 성능 차이가 너무 크기 때문이다. 그림 8과 9는 그림 7의 공간질의 영역 (a), (b), (c), (d)를 수행하였을 때, 각각 $T_{sending}$ 과 $T_{receiving}$ 의

실험 결과를 보여준다.

먼저, 그림 8을 보면 센서노드의 수가 증가함에 따라 SPIX와 CPS 방법의 $T_{sending}$ 값이 비례적으로 증가함을 볼 수 있다. 특히, (a)와 같이 베이스스테이션으로부터 가장 멀리 떨어진 질의 영역의 경우에는 (c)에 비하여 $T_{sending}$ 값이 더욱 크게 증가함을 볼 수 있다. 이에 비하여 GR-tree 방법은 (a), (b), (c), (d)의 질의 영역에 관계없이 센서노드의 수가 증가함에 따라 $T_{sending}$ 값이 큰 차이없이 매우 완만하게 증가하고 있음을 볼 수 있다. 이는 GR-tree 방법이 다른 방법들에 비하여 센서노드들에 대한 분산 공간색인을 효율적으로 구축함으로써 센서노드에 대한 공간 필터링을 효율적으로 수행하기 때문이라고 할 수 있다. 그림 9의 $T_{receiving}$ 에서도 센서노드 수의 증가와 관계없이 GR-tree 방법이 다른 방법들에 비하여 좋은 성능을 보여주고 있음을 알 수 있다. 이는 GR-tree 방법이 다른 방법들에 비하여 다수 센서노드의 센싱결과를 베이스스테이션까지 전송하기 위한 색인 트리의 Depth가 낮고 Multiple Path의 수가 작기 때문이다. 물론, 그림 9(c)와 같이 가까운 질의 영역의 경우에는 색인 트리의 Depth와 Path에서 큰 차이를 얻기 힘들기 때문에 GR-tree 방법과 다른 방법 간에 성능차이가 거의 존재하지 않음을 볼 수 있다. 끝으로, 본 실험에서 검색된 센서노드 수는 2,000 개의 노드가 주어졌을 때 107~130개, 10,000 개의 노드가 주어졌을 때 602~646개, 20,000개의 노드가 주어졌을 때 1,219~1,308개로, 전체 노드 수 대비 5.5%~6.5% 수준의 센서노드가 검색결과로 선택되었다.

두 번째 실험에서는 공간질의 영역의 크기가 다양하게 변경될 때, GR-tree, SPIX(Area), SPIX (Perimeter), CPS, 그리고 분산 공간색인을 이용하지 않는 Centralized 방법의 검색 성능을 비교 분석하였다. 여기서, 보다 공정한 비교분석을 위하여 Centralized 방법은 모든 센서노드의 정보를 수집하기 위한 메시지를 전송하는 과정과 센싱결과를 수집하는 과정에서 무선통신 횟수가 기하급수적으로 증가하는 Flooding 기반의 라우팅 대신에, 트리 기반의 라우팅을 이용한다고 가정한다. 구체적으로, GR-tree의 공간 필터링 기능은 이용하지 않고, 메시지 전송 및 센싱결과 수집은 GR-tree를 이용한다고 가정하였다. 본 실험에서는 2,000x2,000 meters의 전체 영역에 대하여 센서노드의 수는 10,000개로 고정

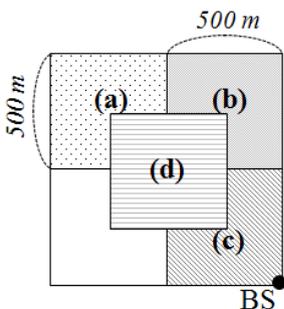


그림 7. 다양한 위치의 공간질의 영역

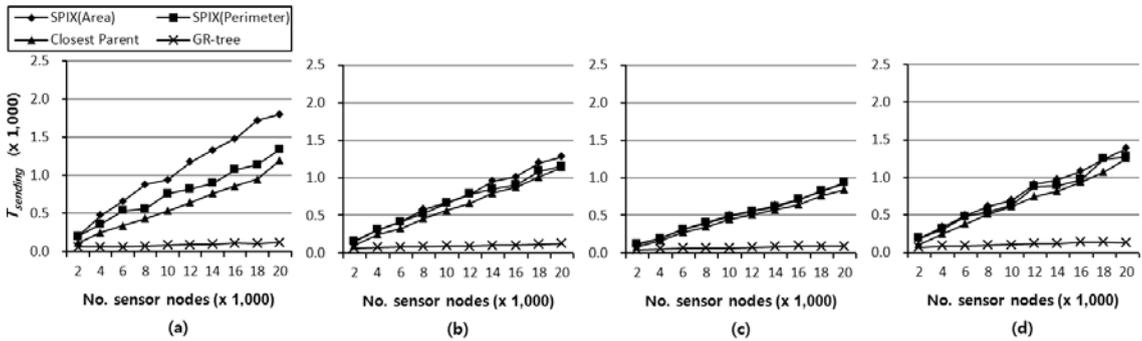


그림 8. 센서노드 수 변화에 따른 GR-tree, SPIX, Closest Parent Selection 방법의 공간질의 전송에 소요되는 무선통신 횟수 ($T_{sending}$) 비교

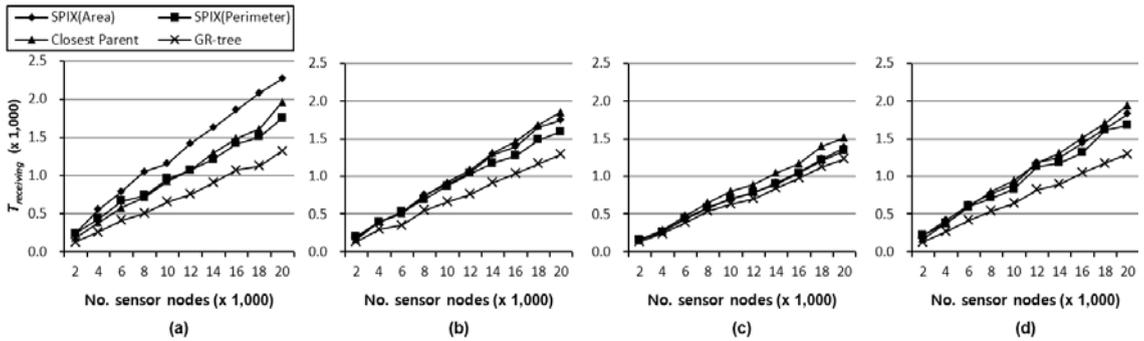


그림 9. 센서노드 수 변화에 따른 GR-tree, SPIX, Closest Parent Selection 방법의 공간질의 수행 후의 결과 수집에 소요되는 무선통신 횟수 ($T_{receiving}$) 비교

시키고, 공간질의 영역의 크기는 200 x 200 meters로부터 2,000 x 2,000 meters의 전체 영역까지 증가시키면서 성능을 측정하였다. 본 실험에서도 공간질의 영역의 위치에 따른 GR-tree 방법의 성능을 분석하기 위하여 그림 10(a)와 같이 베이스스테이션에서 가장 먼 지점부터 질의 영역이 점차적으로 전체 영역으로 확장되는 경우와 그림 10(b)와 같이 베이스스테이션 지점으로부터 질의 영역이 전체 영역으로 확장되는 경우에 대하여 실험을 수행하였다. 그

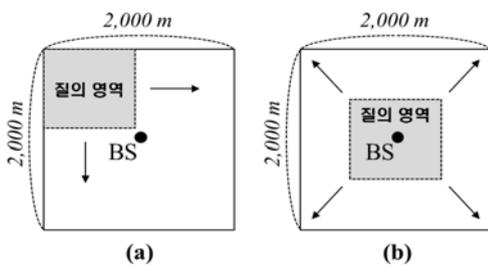


그림 10. 공간질의 영역 변화

림 11과 12는 그림 10(a)와 10(b)에 대한 실험 수행 결과로, $T_{sending}$, $T_{receiving}$ 및 T_{total} ($=T_{sending} + T_{receiving}$)의 값을 각각 보여준다.

먼저, 그림 11과 12는 무선통신 횟수에서의 차이가 존재하지만 그림 10의 질의 영역 위치와 관계없이 유사한 패턴의 결과를 보여주고 있음을 알 수 있다. $T_{sending}$ 을 보면 질의 영역이 800 x 800 meters 이하일 때는 공간색인을 이용한 대부분의 방법들이 Centralized 방법보다 좋은 성능을 보여주고 있다. 이는 분산 공간색인에 의한 공간 필터링의 효과가 큰 힘을 발휘하기 때문이라고 할 수 있다. 그러나, 질의 영역이 1,000x1,000 meters 이상이 되면 GR-tree를 제외한 다른 방법들은 Centralized 방법보다 성능이 떨어짐을 알 수 있다. 이는 질의 영역의 크기가 너무 커지는 경우에는 정해진 트리를 따라서 단순히 센싱정보 수집을 위한 메시지를 브로드캐스팅하는 것이 공간여과를 이용하는 것보다 효율적임을 의미하는 것이라고 할 수 있다. GR-tree의 경우

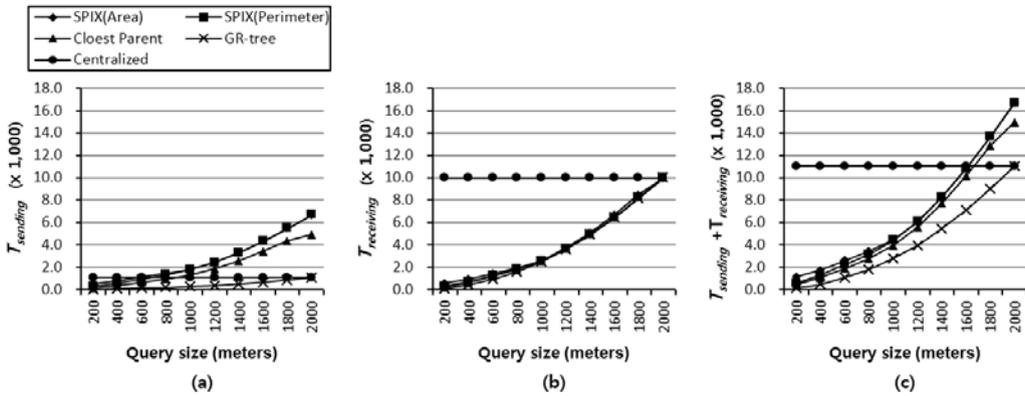


그림 11. 그림 10(a)의 공간질의 영역 크기 변화에 따른 GR-tree, SPIX, Closest Parent Selection, Centralized 방법의 성능 비교: (a) $T_{sending}$, (b) $T_{receiving}$, (c) T_{total} (= $T_{sending}$ + $T_{receiving}$)

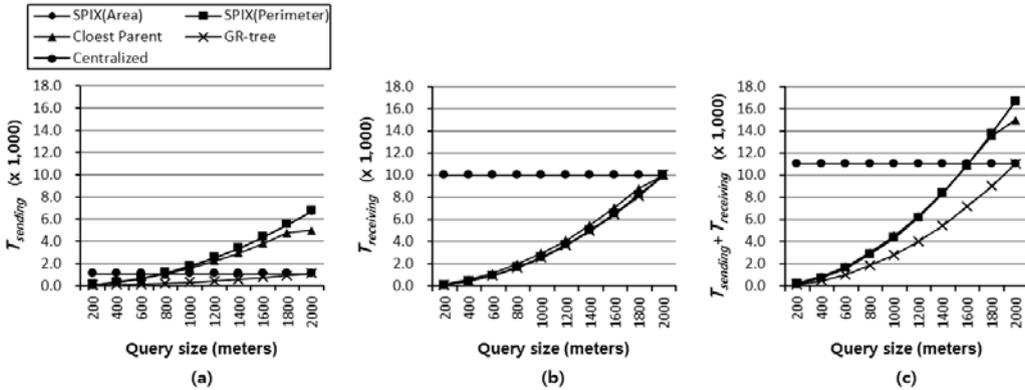


그림 12. 그림 10(b)의 공간질의 영역 크기 변화에 따른 GR-tree, SPIX, Closest Parent Selection, Centralized 방법의 성능 비교: (a) $T_{sending}$, (b) $T_{receiving}$, (c) T_{total} (= $T_{sending}$ + $T_{receiving}$)

는 공간여과의 효율성이 다른 방법에 비하여 뛰어나기 때문에 질의 영역이 2,000×2,000 meters가 되기 전까지는 Centralized 방법보다 좋은 성능을 보여주고 있다. 그러나, 질의 영역이 전체 영역이 되는 경우에는 GR-tree 방법도 Centralized와 동일한 성능을 보여주게 되는데, 이는 앞에서 설명한 바와 같이 Centralized 방법이 내부적으로 GR-tree의 트리 기반 라우팅을 이용하고 있기 때문이다. $T_{receiving}$ 에서는 당연히 공간색인을 이용하는 방법들이 공간적으로 여과되어 선택된 센서노드의 정보만을 서버로 전송하기 때문에 전체 센서노드의 정보를 서버로 전송해야 하는 Centralized 방법에 비하여 좋은 성능을 보여주게 된다. 공간색인을 이용하는 방법 중에서도, GR-tree 방법이 SPIX 및 CPS 방법에 비하여 미미

하게 좋은 성능을 보여준다. 이는 GR-tree 방법이 다른 방법들에 비하여 센싱결과를 전송하기 위한 색인 트리의 Depth가 낮고 Path의 수가 작기 때문이다. 끝으로, T_{total} 을 살펴보면 질의 영역 크기가 1,600×1,600 meters 까지는 GR-tree를 포함한 공간색인을 이용한 방법들이 Centralized 방법에 비하여 우수한 성능을 보여주고 있음을 알 수 있다. 특히, GR-tree의 경우는 질의 영역이 전체 영역이 되는 시점의 이전까지는 Centralized 방법을 포함한 다른 방법들에 비하여 가장 좋은 성능을 보여주고 있다. 본 실험에서 질의 영역 증가(200 ~ 2,000 meters)에 따른 검색된 센서노드의 수는 100 여개에서 10,000 여개로, 전체 노드 수 대비 1%에서 100%까지 이르렀다.

5. 결론

본 논문에서는 센서 네트워크에서 동작 가능한 에너지 효율적인 분산 공간색인기법으로 GR-tree 방법을 제안하였다. 제안된 GR-tree 방법은 기존의 R-tree 방법과 마찬가지로 MBR 기반 클러스터링을 이용하여 분산 공간색인 트리를 구성하는데, 공간색인의 효과를 최대화하기 위하여 클러스터링된 MBR 들 간의 겹침을 최소화하면서 동시에 센서노드들 간의 무선 라우팅도 보장할 수 있는 특징을 가지고 있다.

실험 검증에서는 Random 하게 분포된 센서노드들에 대하여 다양한 실험을 통하여 GR-tree 기법과 기존의 SPIX, CPS 기반의 공간색인기법 그리고 공간색인을 이용하지 않는 Centralized 방법과의 성능을 비교 분석하였다. 그리고 실험 결과에서, GR-tree 방법은 센서노드 수의 많고 적음에 관계없이 또한 길의 영역의 크기에 관계없이 좋은 성능을 보여줄 수 있었다. 특히, 센서노드가 Skewed 하게 분포된다고 할지라도, 균형 트리를 생성하지 않는 GR-tree, SPIX, CPS의 분산 공간색인 환경에서는 GR-tree가 가장 좋은 성능을 보여줄 수 있다. 그러나, GR-tree 기법은 트리 기반의 라우팅을 이용하기 때문에 베이스스테이션에 가까이에 위치한 부모 센서노드일수록 에너지 소모가 커지게 되는 위험성을 가지고 있는데, 센서노드가 Skewed 하게 분포되는 경우, 자식 센서노드의 수가 크게 증가하기 때문에 부모 센서의 에너지 소모가 더욱 커지는 문제점을 동시에 가지고 있다. 또한, 새로운 센서노드가 추가되거나 기존의 센서노드가 삭제될 때 GR-tree를 최적화하여 재구성하는데 많은 무선통신 비용이 소모되는 단점도 가지고 있다.

이에, 향후 연구로는 센서노드의 동적인 변동 상황에 쉽게 대처할 수 있고, 또한 에너지 소모가 큰 부모 센서노드를 동적으로 인접한 다른 센서노드로 대체할 수 있는 동적 GR-tree 구성 방법에 대하여 살펴보고자 한다.

참고 문헌

- [1] P. Bonnet, J. Gehrke and P. Seshadri, 2001, "Towards Sensor Database Systems," Proc. 2nd Int'l Conf. Mobile Data Management, LNCS 1987, pp 3 - 14.
- [2] T. Brinkhoff, H. Kriegel, R. Schneider and B. Seeger, 1990, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD, pp. 322-331.
- [3] M. Demirbas and H.Ferhatosmanoglu, 2003, "Peer-to-Peer Spatial Queries in Sensor Networks," Proc. 3rd Int'l Conference on Peer-to-Peer Computing, pp. 32-39.
- [4] M. Demirbas and X. Lu, 2007, "Distributed Quad-Tree for Spatial Querying in Wireless Sensor Networks," Proc. IEEE Int'l Conference on Communications, pp. 3325-3332.
- [5] A. Guttman, 1984, "R-trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD, pp. 47-57.
- [6] M. S. Kim, J. W. Kim and M. H. Kim, 2008, "Semijoin- Based Spatial Join Processing in Multiple Sensor Networks," ETRI Journal, vol. 30. no. 6, pp. 853-855.
- [7] M. S. Kim, J. W. Kim and M. H. Kim, 2010, "Hybrid Spatial Query Processing between a Server and a Wireless Sensor Network," IEICE. TRANS. INF. & SYST., vol. E93-D, no. 8, pp. 2306-2310.
- [8] C. K. Lee, B. Zheng, W. Lee and J. Winter, 2007, "Materialized In-Network View for Spatial Aggregation Queries in Wireless Sensor Network," ISPRS Journal of Photogrammetry & Remote Sensing, vol. 62, no. 5, pp. 382-402.
- [9] S. Madden, M. J. Franklin and J.M. Hellerstein, 2005, "TinyDB: An Acquisitional Query Processing System for Sensor Networks," ACM TODS, vol. 30, no. 1, pp. 122 - 173.
- [10] A. Meka and A. Singh, 2005, "DIST: A Distributed Spatio-temporal Index Structure for Sensor Networks," Proc. ACM CIKM, pp. 139-146.
- [11] K. Park, B. Lee and R. Elmasri, 2007, "Energy Efficient Spatial Query Processing in Wireless Sensor Networks," Proc. 21st Int'l Conference on Advanced Information Networking and Applications Workshops, pp. 719-724.

- [12] M. Sharifzadeh and C.Shahabi, 2004, "Supporting Spatial Aggregation in Sensor Network Databases," Proc. 12th ACM Int'l Workshop on Geographic Information Systems, pp.166-175.
- [13] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen and M. Welsh, 2004, "Simulating the Power Consumption of Large-Scale Sensor Network applications," Proc. ACM Sensys, pp. 188-200.
- [14] A. Soheili, V. Kalogeraki and D. Gunopulos, 2005, "Spatial Queries in Sensor Networks," Proc. 14th ACM Int'l Workshop on Geographic Information Systems, pp. 61-70.
- [15] M. L. Yiu, N. Mamoulis and S.Bakiras, 2007, "Retrieval of Spatial Join Pattern Instances from Sensor Networks," 19th Int'l Conference on Scientific and Statistical Database Management. pp. 25-34.
- [16] 김정준, 신인수, 이기영, 한기준, 2011, "무선 센서 네트워크에서 효율적인 집계 질의 처리", 한국공간정보학회지, 제19권, 제3호, pp. 95-106.
- [17] 김호, 백성하, 이동욱, 김경배, 배혜영, 2010, "GeoSensor 환경에서 공간질의 정확도 향상을 위한 선-필터링을 이용한 후-부하제한 기법", 한국공간정보학회지, 제12권, 제1호, pp. 18-27.
- [18] 홍승태, 윤민, 장재우, 2010, "지오센서 네트워크에서 에너지 효율성을 지원하는 마이트코드 기반 데이터 서비스 미들웨어 구현, 한국공간정보학회지, 제18권, 제4호, pp. 75-88.

논문접수 : 2011.07.28
수정일 : 2011.10.02
심사완료 : 2011.10.20



김민수

1994년 부산대학교 전산학과 이학사
1996년 부산대학교 대학원 전산학과
이학석사

2010년 KAIST 전산학과 공학박사
2010년~현재 UST 겸임교수

1996년~현재 한국전자통신연구원 책임연구원
관심분야는 Spatial DBMS, Sensor Network DBMS,
LBS



장인성

1999년 부산대학교 전산학과 이학사
2001년 부산대학교 대학원 전산학과
이학석사

2008년 부산대학교 대학원 전산학과
박사수료

2001년~현재 한국전자통신연구원 선임연구원
관심분야는 Spatial Analysis, Navigation, Ubiquitous
GIS, LBS