Journal of Korea Spatial Information Society Vol.21, No.4 : 7-14, August 2013 ISSN 2287-9242(Print)
http://dx.doi.org/10.12672/ksis.2013.21.4.007 ISSN 2287-9250(Online)

도로 네트워크에서 효율적인 범위 검색 기법

An Efficient Range Search Technique in Road Networks

박 춘 걸* 김 정 준** 박 지 웅*** 한 기 준****
Chun Geol Park Jeong Joon Kim Ji Woong Park Ki Joon Han

요 약 최근 도로 네트워크 환경에서 범위 검색에 관한 연구가 활발히 진행되고 있다. 그러나 기존의 대표적인 범위 검색 기법들은 POI(Point of Interest)의 개수가 많을수록 저장 공간이 증가하거나 비효율적인 검색 과정으로 인해 검색 시간이 오래 걸리는 단점이 있다. 따라서 본 논문에서는 이러한 기존 범위 검색 기법들의 문제점을 해결하기 위해 QRMP(QR-tree using Middle Point)를 이용하는 범위 검색 기법을 제시하였다. 그리고 QRMP의 전체 저장 공간 크기를 구하는 수식을 산출하고, 또한 실제 도로 네트워크와 POI 데이터를 이용한 실험을 통해 본 논문에서 제안하는 범위 검색 기법의 우수성을 입증하였다.

키워드 : 도로 네트워크, POI, 범위 검색, QRMP

Abstract Recently, R&D(Research and Development) is processing actively on range search in the road network environments. However, the existing representative range search techniques have shortcomings in that the greater the number of POI's, the more increased storage space or the more increased search time due to inefficient search process. Accordingly, In this paper, we proposed a range search technique using QRMP(QR-tree using Middle Point) to solve the problems of conventional range search techniques. In addition, we made a formula to obtain the total size of the storage space for QRMP and proved the excellence of the range search technique proposed in this paper through the experiment using actual road networks and POI data.

Keywords : Road Networks, POI, Range Search, QRMP

1. 서 론

대용량의 공간 데이터를 이용하는 위치 기반 서비스(LBS, Location-based Service)를 효과적으로 지원하기 위해서는 사용자의 현재 위치를 기반으로 하여 정보/데이터를 탐색하는 위치 기반 질의(Location-based Query)의 효율적 수행이 필수적이다[1,8]. 위치 기반 질의 중 대표적인 범위 검색은 질의 점으로부터 일정 네트워크 거리 안에 존재하는 POI들을 검색하는 것이다[5,6]. 여기서 네트워크 거리는 질의 객체가 도로를 따라 POI에 도달할 때까지 경유한 이동 거리를 의미한다.

도로 네트워크에서 범위 검색 기법에 대한 대표적인 예로는 RNE 기법[7], Signature 기법[3], S-Grid 기법[1] 등이 있다. 그러나 이러한 기존의 범위 검색 기법은 POI의 개수가 많을수록 저장 공간이 증가하거나 효율적이지 못한 검색 과정으로 인해 검색 시간이 오래 걸리는 단점이 있다[11]. 특히, 범위 검색에서 검색 범위의 값이 클수록 후보 POI가 많아지기 때문에 검색 성능이 현저히 떨어진다.

따라서, 본 논문에서는 도로 네트워크 환경에서 이러한 기존 범위 검색 기법의 문제점을 해결하고 대용량 공간 데이터의 보다 효율적인 처리를 지원하기 위해 QRMP(QR-tree using Middle Point)[9,10]를 이용하는 효율적인 범위 검색 기법을 제시하였다. QRMP는 전체 네트워크 공간을 으로 분할하여 Quad-tree를

† This work (Grants No. C0027296) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

* Chun-Geol Park, Consultant, EN-CORE GROUP. cgpark@en_core.com
** Jeong-Joon Kim, Assistant Professor, Dept. of Computer Science and Engineering, Konkuk University. jjkim9@db.konkuk.ac.kr
*** Ji-Woong Park, Associate Professor, Dept. of Computer Information System, Gyeonggi College of Science and Technology. jiwpark@gtec.ac.kr
**** Ki-Joon Han, Professor, Dept. of Computer Science and Engineering, Konkuk University. kjhan@db.konkuk.ac.kr (Corresponding Author)

생성하고, 들의 인접 정보를 관리하기 위해 MP (Middle Point)를 설정하며, 또한 분할된 에 포함된 도로 네트워크의 링크 정보를 R-tree에 저장한다[4]. 즉, QRMP에서는 내에 포함된 POI 개수와 MP 개수를 기준으로 분할을 수행하고, 동일 내의 POI와 MP, 그리고 노드 사이의 네트워크 거리를 저장한다. QRMP는 이러한 정보들을 관리함으로써 범위 검색을 효율적으로 지원할 수 있다.

마지막으로 본 논문에서는 QRMP의 전체 저장 공간 크기를 구하는 수식을 산출하고, 또한 실제 도로 네트워크 데이터와 POI 데이터를 이용해 기존 범위 검색 기법들에 대해 다양한 성능 평가를 수행하여 본 논문에서 제안한 범위 검색 기법의 우수함을 입증하였다.

2. 관련연구

2.1 QRMP(QR-tree using Middle Point)

QRMP는 Quad-tree, Component, LN Component, MP Component, Connection Component, Cell R-tree, MP R-tree로 구성된다[9,10]. Fig.1은 QRMP의 전체 구조를 보여준다.

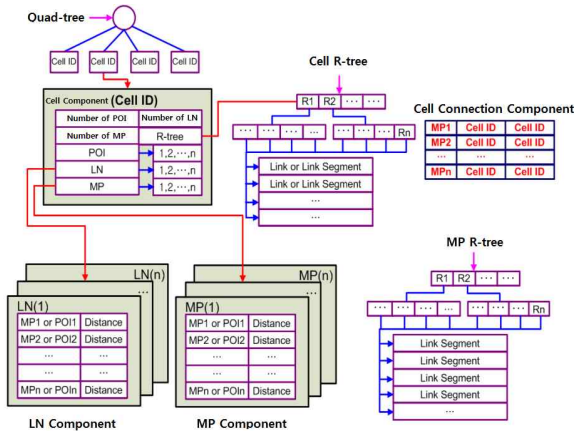

Figure 1. Structure of QRMP

Fig. 1에서와 같이 Quad-tree에서 루트 노드는 영역 정보, 자녀 노드를 가리키는 포인터(루트 노드가 단말 노드일 경우 Cell Component를 가리키는 포인터)를 갖고 있다. 또한, 중간 노드는 영역 정보, 부모 노드와 자녀 노드를 가리키는 포인터를 갖고 있으며, 단말 노드는 영역 정보, 부모 노드를 가리키는 포인터와 Cell Component를 가리키는 포인터를 갖고 있다. 여기서 단말 노드의 영역은 하나의 cell과 매칭이 된다. Cell Component에는 POI의 개수, LN의 개수, MP의 개수, Cell R-tree를 가리키는 포인터를 저장하고, 또한 현재 Cell Component에 저장된 POI들의 아이디, LN들의 아이디, 그리고 MP들의 아이디를 저장한다. LN Component에는 LN과 같은 Cell Component에 저장된 MP와 POI의 아이디를 저장하고 해당 LN부터 MP 또는 POI까지의 네트워크 거리를 저장한다. 그리고 MP Component에는 MP와 같은 Cell Component에 저장된 MP와 POI의 아이디를 저장하고, 또한 해당 MP부터 기타 MP 또는 POI까지의 네트워크 거리를 저장한다. Cell Connection Component에는 MP를 포함하는 cell 아이디를 저장한다. Cell R-tree는 현재 cell에 링크 노드가 포함된 링크 또는 링크 세그먼트를 저장하기 위한 R-tree이고, MP R-tree는 MP를 포함하는 링크 세그먼트를 저장하기 위한 R-tree이다.

QRMP 생성을 위하여 주어진 입력 데이터로 부터 우선 링크를 삽입한 후에 POI를 삽입한다. 링크 데이터를 입력하기 전 우선 입력할 링크 데이터의 x_min, x_max, y_min, y_max값을 이용하여 초기영역을 설정하고 QRMP 루트 노드와 Cell Component를 생성한다. 그런 다음에 링크를 삽입하게 된다. 링크를 삽입하고 각 링크에 포함된 링크 노드들을 Cell Component에 저장하고, 링크를 Cell R-tree에 저장한다. 모든 링크에 대하여 삽입을 완료한 후 POI 삽입을 수행한다. POI의 삽입은 POI를 포함하는 링크 또는 링크 세그먼트가 위치한 cell의 Cell Component에 대해서 수행된다. POI를 삽입 시 특정 cell의 POI 개수와 MP 개수의 합이 분할 기준을 초과하면 해당 cell에 대하여 cell 분할을 수행한다.

특히 cell 분할을 수행 시 우선 현재 분할되는 cell 영역에 대하여 사분할을 수행하고 분할된 각 cell에 대하여 Cell Component를 생성한다. 다음 cell 분할 수행 전의 Cell R-tree에 저장된 링크 또는 링크 세그먼트를 분할된 각 cell의 Cell R-tree에 삽입한다. 모든 링크와 POI를 삽입 완료한 후 각 cell의 Cell Component와 cell R-tree에 저장된 링크 또는 링크 세그먼트를 이용하여 모든 cell의 LN Component와 MP Component를 생성한다. QRMP에서 네트워크 거리 계산은 다익스트라 알고리즘을 사용하였다.

2.2 범위 검색 기법

RNE(Range Network Expansion)는 라인 스트링을 하나씩 확장하여 범위 검색을 수행하는 방식이다[7]. RNE는 라인 스트링과 POI를 각각 별도의 2차원 R-tree에 저장하고 한 라인 스트링과 다른 라인 스트링과의 관계를 링크드 리스트로 관리한다. RNE에서는 범위 검색에서 질의 점이 주어졌을 때 우선 질의 점이 포함된 라인 스트링을 찾는다. 그후, 해당 라인

스트링으로 시작하여 인접 라인 스트링을 하나씩 확장해 가며 해당 라인 스트링에 POI가 있는지의 여부를 POI가 저장되어 있는 R-tree를 이용하여 검색한다. 위의 과정을 사용자가 요구한 범위까지 반복적으로 라인 스트링을 확장한다. RNE는 라인 스트링과 POI를 각 R-tree에 저장하기 때문에 저장 공간이 작다는 장점을 가지고 있지만, 라인 스트링을 확장할 때마다 라인 스트링상에 POI 유무를 검사하기 위하여 POI를 저장한 R-tree를 통하여 매번 검색하기 때문에 검색 성능이 떨어진다.

Signature는 Distance Signature과 Backtracking Link를 이용하여 범위 검색을 수행하는 방식이다[3]. Distance Signature에는 하나의 노드로부터 POI까지의 네트워크 거리를 Distance Category의 매칭 정보를 이용하여 요약 정보로 변환하여 저장하고, Backtracking Link에는 하나의 노드로부터 POI에 도달하기 위하여 경유하여할 인접한 노드의 정보를 저장한다. Signature에서는 범위 검색 시 질의 점 q와 검색할 범위가 주어졌을 때 우선 q가 위치한 노드를 검색하고, 해당 노드의 Distance Signature에서 검색할 범위보다 작거나 같은 요약 정보 값을 갖는 POI들을 후보 집합으로 설정한다. 그리고, 각 후보 POI에 대하여 Adjacency list를 이용하여 실제 네트워크 거리를 계산하여 지정한 검색 범위 보다 같거나 작은 POI를 결과로 반환한다. Signature은 모든 노드로부터 모든 POI까지의 네트워크 거리 정보를 저장하기 때문에 POI의 개수가 많을수록 저장 공간의 점유양이 현저히 증가하게 된다. 또한 각 노드로부터 POI까지의 네트워크 거리를 계산 시 네트워크 거리 요약 정보를 이용하므로 검색 범위가 클수록 불필요한 POI를 많이 검색하기 때문에 검색 성능이 좋지 않은 단점이 있다.

S-GRID(Scalable Grid)는 도로 네트워크 공간을 고정 크기의 그리드 cell로 분할하고 분할된 그리드 cell 정보를 이용하여 범위 검색을 수행하는 방식이다[1]. 우선, 주어진 도로 네트워크 공간을 고정 크기의 그리드 cell로 분할하고, 라인 스트링의 두 끝점이 서로 다른 그리드 cell에 포함될 때 해당 라인 스트링의 중심점을 경계점(Border Point)으로 설정한다. 범위 검색 시 질의 점이 주어졌을 때 S-GRID는 우선 질의 점이 포함된 라인 스트링을 찾는다. 그 후, 해당 라인 스트링으로 시작하여 인접 라인 스트링을 하나씩 확장해 가며 해당 라인 스트링에 POI가 있는지의 여부를 POI가 저장되어 있는 R-tree를 이용하여 검색한다. 현재 검색하는 그리드 cell 내에 POI가 없거나 그리드 cell 내에 포함된 모든 POI에 대한 검색을 마쳤을 경우 현재 cell에 대한 검색을 종료하고 경계점을 이용하여 인접 cell을 검색한다. 이와 같은 과정을 사용자가 요구한 검색 범위까지 반복적으로 라인 스트링을 확장하여 나간다. S-GRID는 라인 스트링을 확장할 때마다 라인 스트링에 POI가 존재하는지 유무를 판단하기 위해 POI를 저장한 R-tree를 검색하기 때문에 검색 성능이 떨어진다.

3. QRMP를 이용한 범위 검색 기법

본 논문에서 제시하는 QRMP를 이용한 범위 검색 알고리즘은 검색 버퍼, MP 버퍼를 사용한다. Fig. 2는 검색 버퍼의 구조를 보여 준다.

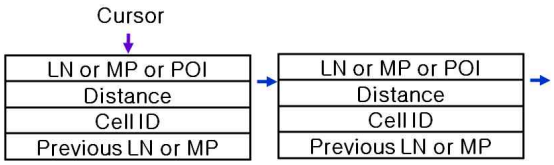

Figure 2. Search Buffer

Fig. 2에서 보는 바와 같이 검색 버퍼에는 LN, MP, POI를 저장하고, 질의 점에서 LN, MP, POI까지의 네트워크 거리를 저장하는 Distance, 현재 LN, MP, POI가 위치한 cell 아이디, 그리고 현재 LN, MP, POI에 도달하기 위하여 경유한 바로 전의 LN 또는 MP를 저장한다. Cursor는 검색 버퍼 내의 현재 검사할 LN, MP, POI 중 하나를 가리키고 초기에는 검색 버퍼의 첫 번째 위치를 가리킨다. MP 버퍼는 검색 버퍼에서 삭제된 MP를 저장함으로써 MP 아이디, MP가 포함되는 cell 아이디, 그리고 현재 MP를 도달하기 위하여 경유한 바로 전의 LN 아이디 또는 MP 아이디를 저장한다. 범위 검색 시 우선 질의 점 q로부터 질의 점이 포함된 링크 또는 링크 세그먼트의 두 끝점까지의 네트워크 거리를 계산한 후 네트워크 거리 오름차순으로 LN, MP를 검색 버퍼에 삽입한다. Fig. 3은 주어진 왼쪽 그림에 대해서 LN3, MP1을 검색 버퍼에 삽입한 예를 보여준다.

Fig. 3에서와 같이 검색 버퍼에 네트워크 거리 오름차순으로 LN3과 MP1이 삽입된다. 이 예의 검색 버퍼에는 질의 점으로부터 LN3까지의 네트워크 거리는 1이고, MP1까지의 네트워크 거리는 2이며, LN3과 MP1이 모두 cell 11에 위치하고 있으며, 또한 이들을 도달하기 위해 다른 LN 또는 MP를 경유하지 않으므로 NULL이 저장된다. 그리고, 검색 버퍼에서 MP가 삭제되지 않았기 때문에 MP 버퍼는 NULL로 설정된다.

다음 검색 버퍼에서 Cursor가 가리키고 있는 데이터를 확인한다. Cursor가 LN을 가리킬 경우 검색 버퍼에

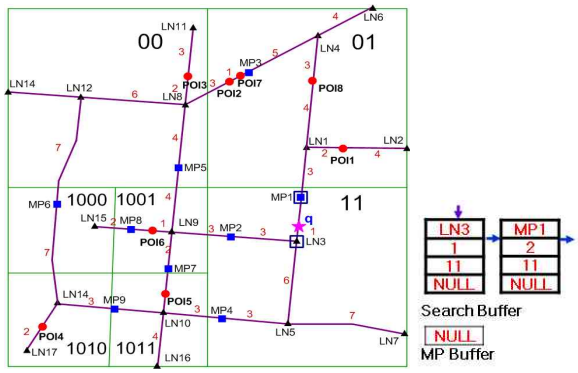

Figure 3. Example of LN3, MP1 Insertion on Search Buffer

서 LN을 삭제하고 질의 점 q로부터 LN Component에 저장된 POI, MP까지의 네트워크 거리를 계산하여 네트워크 거리 오름차순으로 POI와 MP를 검색 버퍼에 삽입한다. 검색 버퍼에서 Cursor가 MP를 가리킬 경우 MP 버퍼에 동일한 MP가 존재하면 검색 버퍼에서 MP를 삭제하고, MP 버퍼에 해당 MP가 존재하지 않으면 검색 버퍼에서 MP를 삭제하고 추가적으로 해당 MP를 MP 버퍼에 삽입한다.

또한, Cursor가 가리키는 MP가 MP 버퍼에 존재하지 않으면 질의 점 q로부터 MP Component에 저장된 POI, MP까지의 네트워크 거리를 계산하여 네트워크 거리 오름차순으로 검색 버퍼에 삽입한다. MP Component의 MP와 POI를 검색 버퍼에 삽입하기 위하여 추가적으로 Cell Connection Component를 이용한다. 검색 버퍼에서 Cursor가 POI를 가리킬 경우 Cursor 이전 위치에 동일 POI가 존재하는지를 확인한다. 동일 POI가 존재하지 않을 경우 Cursor를 다음 위치로 이동한 후 검색을 계속 수행한다. 그리고 Cursor 이전 위치에 동일 POI가 존재할 경우 해당 POI를 검색 버퍼에서 삭제한 후 검색을 계속 수행한다.

이러한 과정을 커서가 NULL을 가리킬 때까지 계속 반복하여 수행한다. Fig. 4는 검색 범위를 10으로 설정하였을 때 범위 검색 결과를 보여준다.

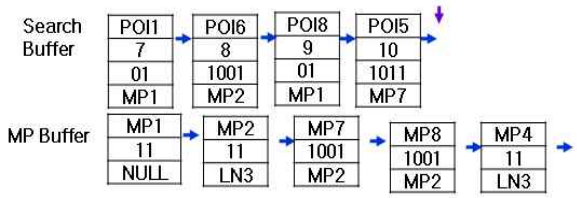

Figure 4. Example of Range Search Result

Fig. 4에서 보는 바와 같이 POI5에 대하여 처리를 완료한 후 커서가 NULL을 가리키기 때문에 검색을 종료하고 결과를 반환한다. 본 예제에서의 결과는 POI1, POI6, POI8, POI5이다. 검색 버퍼, MP 버퍼, 그리고 각 cell의 최단 경로를 이용하여 최종적으로 경로를 안내한다. 따라서 최종 결과는 POI1, 경로 q→LN1→POI1(거리 7); POI6, 경로 q→LN9→POI6(거리 8); POI8, 경로 q→LN1→POI8(거리 9); POI5, q→LN3→LN9→POI5(거리 10)이다. 지금까지 설명한 전체 범위 검색 알고리즘은 Fig. 5와 같다.

```
Function   KNN_Search(Query q, QRMP *root)
Input      q: Query information, root: root information of QRMP
Output     Cursor: information of Cursor in SEARCH_BUFFER
1:    LNK_SEGMENT LS[] = Search_ls(q, QRMP);
2:    SEARCH_BUFFER searbuff =Insert_LN(q, LS[]);
3:    Checking_cursor(q, searbuff){
4:      IF the Cursor is point to POI
5:        IF the same POI exists on previous position of Cursor
      in search_buffer
6:          Delete_POI(Cursor);
7:          Checking_Cursor(q, searbuff);
8:        ELSE
9:          Shift_Cursor(search_buffer, Cursor);
10:         Checking_Cursor(q, searbuff);
11:       END IF
12:   ELSE IF the Cursor is point to LN
13:     Insert_MP_POI(searbuff, Cursor, root);
14:     Delete_LN(searbuff, Cursor);
15:     Checking_Cursor(q, searbuff);
16:   ELSE IF the Cursor is point to MP
17:     IF MP in MP_BUFFER
18:       Delete_MP(searbuff, Cursor);
19:       Checking_Cursor(q, searbuff);
20:     ELSE
21:       Insert_MP_POI(searbuff, Cursor, root);
22:       Insert_MP_Buffer(Cursor, MP_BUFFER);
23:       Delete_MP(search_buffer, * Cursor);
24:       Checking_Cursor(q, searbuff);
25:     END IF
26:   ELSE the Cursor is point to NULL
27:     return Cursor
28:   END IF
29: }
```

Figure 5. Range Search Algorithm

Fig. 5에서 보는 바와 같이 (1~2줄)우선 질의 점이 위치한 링크 또는 링크 세그먼트의 두 끝점을 검색 버퍼에 삽입한 후 Cursor 확인 함수를 호출한다.

Cursor 확인 함수에서 (4~11줄)Cursor가 POI를 가리킬 경우 검색 버퍼에서 Cursor 이전 위치에 동일 POI가 존재하는지 검사한다. 동일 POI가 존재할 경우 검색 버퍼에서 POI를 삭제하고 Cursor 확인 함수를 호출한다. 동일 POI가 검색 버퍼에서 Cursor 이전 위치에 존재하지 않을 경우 Cursor를 다음 위치로 이동시킨 후 Cursor 확인 함수를 호출한다.

(12~15줄)Cursor가 LN을 가리킬 경우 질의 점으로부터 LN Component에 저장된 모든 MP와 POI까지의 네트워크 거리를 계산하여 MP와 POI를 네트워크 거리 오름차순으로 검색 버퍼에 삽입한다. 삽입을 마친 후 LN을 검색 버퍼에서 삭제한 후 Cursor 확인 함수를 호출한다.

(16~25줄)Cursor가 MP를 가리킬 경우 동일 MP가 MP 버퍼에 존재하는지 검사한다. 해당 MP가 MP 버퍼에 존재할 경우 검색 버퍼에서 삭제한 후 Cursor 확인 함수를 호출한다. 동일 MP가 MP 버퍼에 존재하지 않을 경우 MP를 MP 버퍼에 삽입하고, 질의 점으로부터 MP Component에 저장된 모든 MP와 POI까지의 네트워크 거리를 계산하여 MP와 POI를 네트워크 거리 오름차순으로 검색 버퍼에 삽입한다. 그리고 검색 버퍼에서 MP를 삭제한 후 Cursor 확인 함수를 호출한다.

(26~28)Cursor가 NULL을 가리킬 경우 검색을 종료한다.

4. 성능 분석

본 절에서는 QRMP의 전체 저장 용량을 구하는 수식을 유도하고, 실제 도로 네트워크와 POI 데이터를 이용한 성능 평가를 통해 본 논문에서 제안하는 범위 검색 기법의 우수성을 입증한다.

4.1 저장 용량 분석

본본 논문에서 링크는 가로와 세로로 균등하게 분포되어 있고 POI 또한 균등으로 분포되어 있다는 가정 하에 QRMP의 전체 저장 용량을 구하는 수식을 유도하였다. 수식에서 사용될 기호는 Table 1과 같다.

Table 1. Simbols

S	Number of Partition Standard
L	Number of LN Insertion
P	Number of POI Insertion
i	Number of Maximum Partition
D	Size of Data Structure for Storage of Network Distance from One LN(or MP) to POI & MP

먼저, QRMP의 저장 용량을 구하기 위해서는 최대 분할 횟수를 구한다. cell을 i-1번 분할하면 한 개 cell에 저장된 POI와 MP 개수의 합은 S보다 크고 i번 분할하면 한 개 cell에 저장된 POI와 MP의 개수의 합은 S보다 작다. 이것을 부등식으로 표현하면 식 1과 같다.

$$(\frac{1}{4})^i \times P + (\frac{1}{2})^i \times 4\sqrt{L} < S < \left(\frac{1}{4}\right)^{i-1} \times P + \left(\frac{1}{2}\right)^{i-1} \times 4\sqrt{L} \quad (1)$$

식 1로 부터 최종 최대 분할 횟수 i는 식 2를 이용하여 구할 수 있다.

$$i = floor\left(\frac{\sqrt{\left(\frac{2\sqrt{L}}{P}\right)^2 + \frac{S}{P} - \frac{2\sqrt{L}}{P}}}{\log\frac{1}{2}} \right) + 1 \quad (2)$$

여기서 floor는 실수의 소수점 아래 자리 수를 없앤 후 얻은 정수를 의미한다. cell이 분할되면 두 가지 경우가 발생한다. 첫 번째 경우는 단말 cell들의 깊이(Depth)가 같은 경우이고, 두 번째는 단말 cell들의 깊이가 같지 않은 경우이다. 두 경우 중 첫 번째, 단말 cell들의 깊이가 같은 경우가 두 번째 경우보다 저장 용량이 더 크기 때문에 첫 번째 경우만을 고려하여 수식을 표현한다.

다음 QRMP의 저장 용량을 구하기 위해서 cell의 저장 용량을 구한다. cell의 저장 용량은 식 3을 이용하여 구할 수 있다.

$$\text{cell의 개수} \times (\text{MP의 개수} + \text{LN의 개수}) \times (\text{MP의 개수} + \text{POI의 개수}) \times \text{D} \quad (3)$$

식 3에서 사용되는 MP의 개수, LN의 개수, 그리고 POI의 개수는 한 cell 내에 있는 개수들을 의미한다. cell이 분할된 후에는 두 개의 변에만 MP가 있는 cell과 4개의 변에 모두 MP가 있는 cell들이 생성될 수 있다. 두 개의 변에 MP가 존재하는 cell의 개수는 $4\times 2^i - 4$, MP의 개수는 $\left(\frac{1}{2}\right)^i \times 2\sqrt{L}$, LN의 개수는 $\left(\frac{1}{4}\right)^i \times L$, POI의 개수는$\left(\frac{1}{4}\right)^i \times P$ 이다. 따라서, 두 개의 변에 MP가 존재하는 cell의 저장 용량은 식 4와 같다.

$$(4\times 2^i - 4) \times \left[\left(\frac{1}{2}\right)^i \times 2\sqrt{L} + \left(\frac{1}{4}\right)^i \times L\right] \times \left[\left(\frac{1}{2}\right)^i \times 2\sqrt{L} + \left(\frac{1}{4}\right)^i \times P\right] \times \quad (4)$$

4 개의 변에 MP가 존재하는 cell의 개수는 $4^i - [4\times 2^i - 4]$, MP의 개수는 $\left(\frac{1}{2}\right)^i \times 4\sqrt{L}$, LN의 개수는 $\left(\frac{1}{4}\right)^i \times L$, POI의 개수는 $\left(\frac{1}{4}\right)^i \times P$ 이다. 따라서, 4 개의 변에 MP가 존재하는 cell의 저장 용량은 식 5와 같다.

$$[4^i - (4\times 2^i - 4)] \times \left[\left(\frac{1}{4}\right)^i \times L + \left(\frac{1}{2}\right)^i \times 4\sqrt{L}\right] \times \left[\left(\frac{1}{2}\right)^i \times 4\sqrt{L} + \left(\frac{1}{4}\right)^i \times P\right] \times D \quad (5)$$

결론적으로 최종 QRMP의 전체 저장 용량은 식 4와 식 5의 합이 된다. 즉, 위 수식에서 보듯이 QRMP의 전체 저장 용량은 모든 노드로부터 모든 POI까지의 네트워크 거리를 저장하는 것이 아니라 분할된 각 cell의 저장 용량(한 cell 내에서 LN 또는 MP로부터 POI

와 MP까지의 네트워크 거리 저장)의 총합으로 구하기 때문에 전체 저장 용량이 작게 된다.

4.2 성능 평가

성능 평가에 사용된 시스템의 하드웨어 사양은 Intel Core 2.4GHz MPU, 2GB RAM, 300GB HDD이며, 운영체제로는 Windows XP를 사용하고, QRMP를 구현하기 위한 언어로는 Visual Studio 2010, MFC를 사용하였다.

그리고 성능 평가를 위해 서울시 네트워크 데이터와 POI 데이터를 사용하였다. 서울시 네트워크는 111,716개의 라인 스트링과 455,984개의 라인 세그먼트로 구성되고, 서울시 POI 데이터는(백화점, 대형마트, 영화관, 우체국, 전철역, 주유소, 편의점, 은행, 약국, 정비소, 병원 등) 27,877개로 구성되어 있다.

본 논문의 모든 성능 평가에서 비교 대상으로는 RNE, Signature, S-GRID를 사용하였다.

4.2.1 인덱스 생성 성능

인덱스 생성 성능 평가에서는 인덱스 생성 크기와 인덱스 생성 시간을 평가하였다. QRMP는 분할 기준 개수 100을 사용하였고, S-GRID는 20X20 cell 분할을 사용하였다. Fig. 6은 인덱스 생성 크기 성능 평가 결과를 보여준다.

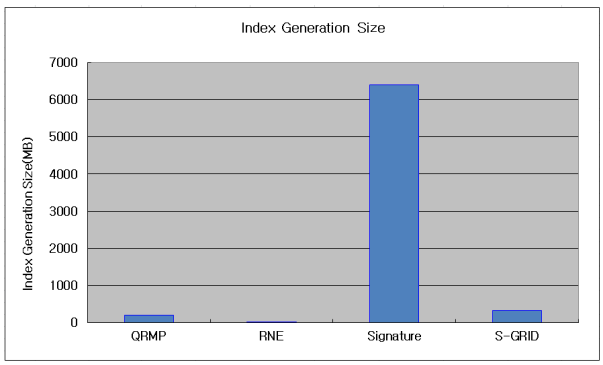

Figure 6. Index Generation Size

Fig. 6에서 보듯이 인덱스 생성 크기 성능 평가를 비교한 결과 QRMP가 RNE 보다 평균 900% 성능이 저하되나 Signature 보다 평균 3,020%, S-GRID 보다 평균 60% 성능이 향상되었다. 이러한 성능 평가의 원인은 다음과 같다. RNE는 각 노드로부터 POI까지의 네트워크 거리를 저장하지 않기 때문에 저장 용량이 QRMP 보다 작고, Signature은 모든 노드로부터 모든 POI까지의 네트워크 거리를 저장하기 때문에 저장 용량이 QRMP 보다 크며, S-GRID는 각 cell에 포함된 네트워크가 크기 때문에 저장 용량이 QRMP 보다 크다. QRMP는 RNE 보다 인덱스 생성 크기는 크지만 POI 개수와 MP 개수를 기반으로 cell을 동적으로 분할하기 때문에 Signature과 S-GRID 보다 인덱스 생성 크기가 작다.

Fig. 7은 인덱스 생성 시간 성능 평가 결과를 보여준다.

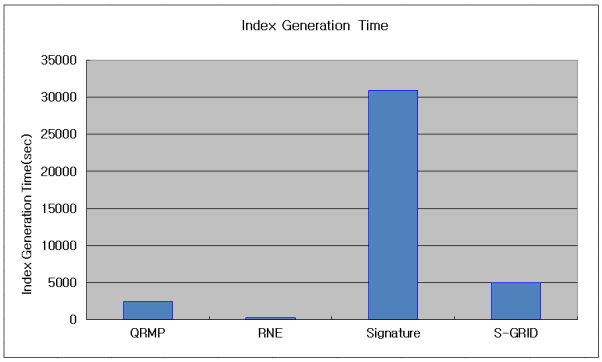

Figure 7. Index Generation Time

Fig. 7에서 보듯이 인덱스 생성 시간 성능 평가를 비교한 결과 QRMP가 RNE 보다 평균 800% 성능이 저하되나 Signature 보다 1,150%, S-GRID 보다 100% 성능이 향상되었다. 이러한 성능 평가의 원인은 다음과 같다. RNE는 노드와 POI간의 네트워크 거리를 계산하지 않기 때문에 QRMP 보다 생성 시간이 작고, Signature은 모든 노드로부터 모든 POI까지의 네트워크 거리를 계산하기 때문에 생성 시간이 QRMP 보다 크며, S-GRID는 각 cell에 포함된 큰 네트워크를 이용하여 각 노드와 경계점간 네트워크 거리를 계산하기 때문에 생성 시간이 QRMP 보다 크다. QRMP는 POI 개수와 MP 개수를 기반으로 cell을 동적으로 분할하여 네트워크 거리를 계산하기 때문에 Signature, S-GRID 보다 인덱스 생성 시간이 작다.

4.2.2 범위 검색 성능

범위 검색 성능 평가에서는 범위 검색 시간을 비교평가하였다. Fig. 8은 R 값에 따른 범위 검색 시간 성능 평가 결과를 보여준다.

Fig. 8에서 보듯이 R 값에 따른 범위 검색 시간 성능 평가를 비교한 결과 QRMP가 RNE 보다 평균 760%, Signature보다 평균 140%, S-GRID보다 평균 470% 성능이 향상되었다. 여기서 R 값은 질의 범위 값을 서울시 사각형 영역의 큰 변의 길이 값으로 나눈 후 100%를 곱하여 얻은 값으로 설정하였다. 범위 검색에서 QRMP는 RNE, Signature, S-GRID 보다 성능이 매우 좋은 결과를 보였다. 이러한 성능 평가의 원인은 다음과 같다. RNE와 S-GRID는 POI 검색 시 라인 스트링

확장 과정에서 POI 유무를 판단하기 위해 매번 POI가 저장된 R-tree를 검색해야 하기 때문에 QRMP보다 검색 성능이 나쁘고, Signature는 각 노드로부터 POI까지의 네트워크 거리를 계산 시 불필요하게 많은 POI를 검색하기 때문에 QRMP 보다 검색 성능이 나쁘다.

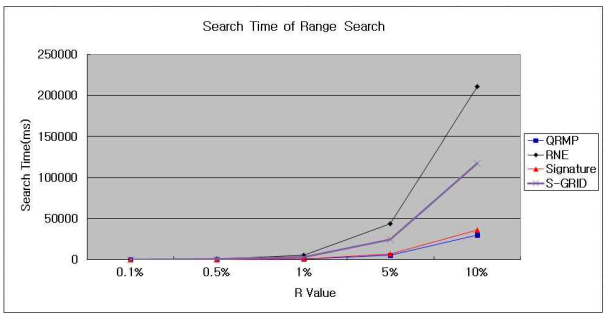

Figure 8. Search Time by R Value

특히, QRMP는 한 개 cell에 저장된 여러 개의 POI를 한 번에 검색 버퍼에 저장하여 검색할 후보 cell을 줄이고, cell들의 인접 정보를 관리하는 MP를 이용하여 MP Component에 저장된 POI를 빠르게 검색하기 때문에 RNE, Signature, S-GRID 보다 검색 성능이 좋다.

5. 결 론

본 논문에서는 도로 네트워크 환경에서 효율적인 범위 검색을 위해 QRMP를 이용하는 범위 검색 기법을 제시하였다. QRMP는 Quad-tree를 이용해 동적으로 도로 네트워크 공간을 cell로 분할하고 범위 검색 시 QRMP에 저장된 POI와 MP간의 미리 계산된 네트워크 거리를 이용하므로 질의 점으로부터 POI 또는 MP까지의 네트워크 거리를 보다 빠르게 구할 수 있다.

본 논문에서 수행한 성능 평가에서 인덱스 생성 크기에서는 QRMP가 RNE 보다 크지만 Signature와 S-GRID 보다 작았고, 인덱스 생성 시간에서는 RNE 보다 컸지만, Signature, S-GRID 보다 빨랐다. 그리고 범위 검색에서도 QRMP는 RNE, Signature, S-GRID 보다 우수한 성능을 보였다.

본 논문에서 제시한 QRMP는 인덱스 생성 크기와 생성 시간에서는 기존 연구들에 비해 다소 성능이 저하되었지만 범위 검색에서는 뛰어난 성능을 보였기 때문에 다양한 위치 기반 서비스(위치 추적 서비스, 안전 및 보안 서비스, 주변 지역 정보 제공 서비스, 광고 및 상거래 응용서비스, 교통 및 항법 서비스 등)에서 활용 가치가 매우 높다. 그 이유는 일반적인 위치 기반 서비스에서 도로나 POI의 갱신 및 추가로 인해 데이터 구축을 다시 하는 것 보다는 초기에 구축된 데이터를 기반으로 범위 검색 질의가 더욱 빈번하게 발생하기 때문이다.

References

[1] Frentzos, E; Gratsias, K; Theodoridis, Y. 2007, Towards the Next Generation of Location-based Services, Proc. of the Int. Workshop on Web and Wireless Geographical Information Systems, LNCS 4857, 202-215.

[2] Huang, X; Jensen, C. S; Lu, H; Šltenis, S. 2007, S-GRID: A Versatile Approach to Efficient Query Processing in Spatial Networks, Proc. of the International Symposium on SSTD, LNCS 4605, 93-111.

[3] Hu, H. B; Lee, D. L; Lee, C. V. S. 2006, Distance Indexing on Road Networks, Proc. of the International Conference on VLDB, 894-905.

[4] Kang, H. K; Shin, I. S; Kim, J. J; Han, K. J. 2010, MR-Tree: A Mapping-based R-Tree for Efficient Spatial Searching, Journal of KSIS, 18(4):109-120.

[5] Lin, H. Y. 2008, Efficient and Compact Indexing Structure for Processing of Spatial Queries in Line-based Databases, Journal of Data and Knowledge Engineering, 64(1):365-380.

[6] Papadias, D; Theodoridis, Y; Stefanakis, E. 1997, Multidimensional Range Query Processing with Spatial Relations, Journal of Geographical Systems, 4(4):343-365.

[7] Papadias, D; Zhang, J; Mamoulis, N; Tao, Y. 2003, Query Processing in Spatial Network Databases, Proc. of the International Conference VLDB, 802-813.

[8] Park, C. G; Kang, H. K; Kim, J. J; Han, K. J. 2009, A Hash-based R-tree for Fast Search of Large Spatial Data, Proc. of the International Conference on EDB, 5-50.

[9] Park, C. G; Kim, J. J; Han, K. J. 2013, Efficient POI Search for Location-based Services in Road Networks, Journal of KIISE: Database, 40 (2):141-151.

[10] Park, C. G. 2012, Efficient POI Search for Location-based Services in Road Networks,

Konkuk University.

[11] Park, C. G; Park, H. H; Kang, H. K; Han, K. J. 2007, Development of an OpenGIS Spatial Interface based on Oracle, Journal of KSIS, 9(2):1-11.

논문접수 : 2013.02.04
수 정 일 : 2013.06.25
심사완료 : 2013.07.22