장애물이 존재하는 검색공간에서 역최대근접질의 처리방법에 관한 연구

선휘준* · 김홍기**

요 약

암호화된 공간데이터베이스와 같은 최근의 여러 응용에서는 질의 기준이 최대근접객체가 되는 객체들을 찾는 역최대근접질의가 자주 발생한다. 실세계의 검색공간에는 강, 호수 그리고 고속도로 등과 같은 다양한 장애물이 존재하며, 이러한 환경에서 검색성능을 높이기 위해서는 장애물을 고려한 검색거리 측도가 반드시 필요하다. 본 연구에서는 장애물이 존재하는 검색공간에서 역최대근접질의 처리를 최적화하기 위한 검색거리 측도들과 질의처리 알고리즘을 제시한다.

The Processing Method for a Reverse Nearest Neighbor Queries in a Search Space with the Presence of Obstacles

Seon Hwi Joon* · Kim Hong Ki**

ABSTRACT

It is occurred frequently the reverse nearest neighbor queries to find objects where a query point can be the nearest neighbor object in recently applications like the encrypted spatial database. In a search space of the real world, however, there are many physical obstacles(e.g., rivers, lakes, highways, etc.). It is necessary the accurate measurement of distances considered the obstacles to increase the retrieval performance such as this circumstance. In this study, we present the algorithm and the measurement of distance to optimize the processing performance of reverse nearest neighbor queries in a search space with the presence of obstacles.

Key words : Reverse nearest neighbor queries, Search distances, Obstacles, Encrypted spatial database

접수일(2017년 5월 23일), 게재확정일(2017년 6월 26일)

* 신경대학교 /ICT융합학과(주저자)
** 동신대학교/융합정보보안학과(교신저자)

1. 서 론

공간데이터베이스에서 시스템의 성능을 향상시키기 위해서는 동적 및 정적환경에서 발생하는 역최대근접질의(reverse nearest neighbor queries)를 효율적으로 처리하는 질의처리 방법이 필요하다[1,2,4,5,6,8,9,11,12].

위치를 기반으로 하는 여러 응용에서는 사용자로부터 수집한 공간 데이터가 인증되지 않은 사용자들에게 유출되는 것을 원하지 않는다. 또한, 질의를 수행하는 동안 자신의 위치가 노출되지 않고 정확한 질의결과가 산출되기를 원한다. 이러한 환경에서는 암호화된 공간 데이터를 기반으로 한 최적화된 역최대근접질의의 처리가 요구된다.

실세계의 검색공간에는 강, 호수 그리고 고속도로 등과 같은 여러 종류의 장애물이 존재하기 때문에 정확한 질의처리를 위해서는 반드시 장애물을 고려한 검색거리 측도를 고려해야 한다[3,7,14].

이러한 장애물을 고려한 객체들 간의 검색거리 측도를 사용함으로써 질의 기준이 최대근접객체가 되는 객체들을 찾는 역최대근접질의 처리비용을 최소화할 수 있다.

본 연구에서는 다차원 검색공간에서 장애물과 객체들의 위치속성을 고려한 역최대근접질의 처리방법을 제안한다. 제안한 방법에서는 역최대근접질의 처리비용을 최소화하기 위한 검색거리인 최소장애거리(minimum obstructed distance : MINODIST)와 최적장애거리(optimized minimum value of all obstructed distances : OMINODIST) 그리고 최대장애거리(maximum value of all obstructed distances : MAXODIST)를 제시한다.

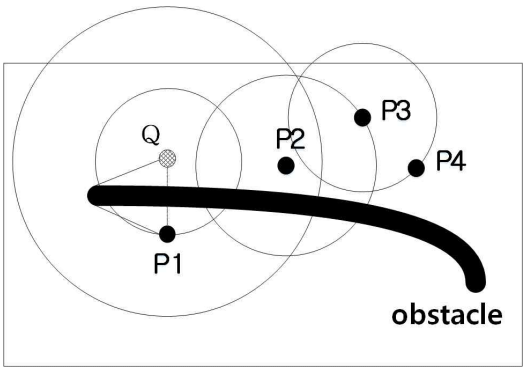
2. 관련연구

실세계의 검색공간 안에는 여러 종류의 장애물이 존재하며, 역최대근접질의의 처리시 이를 고려하지 않으면 올바르지 않은 질의처리 결과로 인해 비효율적인 질의처리 성능을 가져올 수 있다[13].

예를 들면 '광주에서 새로운 주유소를 개점하려고 할 경우 기존의 주유소에 적대적 영향을 주지 않고 고객들을 유치할 수 있는 최적의 위치를 결정한다고 하자. 이 경우 해당위치에서 가장 영향을 주는 경쟁 주유소를 찾아라'와 같은 역최대근접질의의 처리는 빌딩이나 아파트 등과 같은 장애물들을 고려하여 질의결과를 산출해야 한다. 이러한 역최대근접질의의 처리는 연산 및 보조기억장치 접근을 위한 많은 처리 시간이 요구된다. 보조기억장치의 접근시간은 연산 시간에 비해 상대적으로 매우 크기 때문에 역최대근접질의의 검색성능을 높이기 위해서는 보조기억장치의 접근 횟수를 최소로 할 수 있는 질의 처리 방법이 반드시 필요하다.

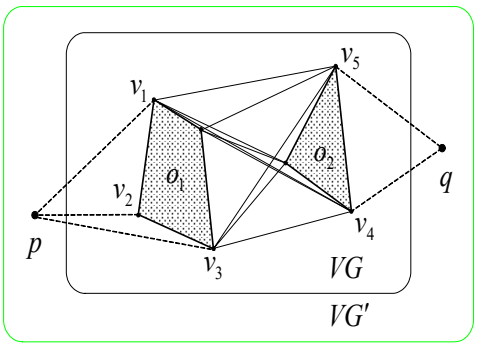
그림 1은 장애물이 있는 검색공간에서 역최대근접질의를 처리하는 예를 나타낸 것이다. 객체 가 포함되고 있는 원안에 있는 모든 객체들은 객체 $P2$의 후보 최대근접객체임을 의미한다. 만약 장애물을 고려하지 않고 유클리드 거리를 측도로 사용하여 역최대근접질의를 처리한다면 $P1$만을 고려하게 된다. 그러나 장애물을 고려한 유클리드 거리가 사용된다면 $P2$도 최대근접거리내에 포함되므로 역최대근접질의의 결과를 정확하게 산출하게 된다. 예에서는 Q의 역최대근접객체는 $P1$이 아니라 $P2$가 된다.

이와같이 검색공간에 장애물이 존재한다면 역최대근접질의의 처리시 장애물의 고려는 정확한 질의결과에 주요한 요인이 된다.



(그림 1) 검색공간에서 역최대근접질의

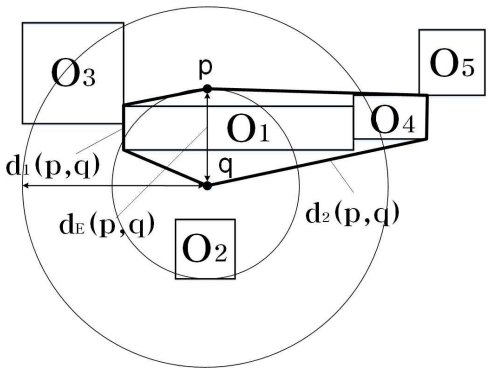
검색공간에서 장애 거리는 일반적인 유클리드 거리 함수를 변경함으로써 구현될 수 있으며 장애 거리는 가시 그래프(visibility graph)를 이용하여 구할 수 있다[3]. 가시 그래프 G (V,E)는 장애물들의 정점 집합 V와 서로 보이는 정점을 연결한 간선 집합 E로 구성된다. VG를 만들기 위해서는 장애물들의 정점을 조사하여 서로 보이는 모든 정점의 집합을 구해야 한다. 이는 두개의 객체가 서로 보이는가를 결정할 수 있는 데이타구조인 BSP-트리를 사용함으로써 해결할 수 있다. 객체 p와 q사이의 장애 거리를 구하기 위해서는 VG의 확장된 가시 그래프인 $VG' = (V', E')$를 생성해야 한다. 가시 그래프 VG에서 V는 장애물들의 정점 집합 V에 객체 p,q의 위치를 포함시킨 것이고, E는 V에 있는 서로 보이는 정점들에 대한 간선 집합이다.



(그림 2) 가시 그래프

그림 2에서는 VG가 두개의 장애물 o ,o_2이 포함된 검색공간에 따른 VG에서 유도됨을 보인다. 객체 p에서 q까지의 장애 거리는 객체 p의 위치에서 시작해서 VG의 정점 ,v_2,v_3 중 하나를 경유하여 v_4 또는 v_5를 거쳐 객체 q의 위치까지의 최소의 값을 갖는 간선 경로이다.

그림3은 장애거리를 계산하는 방법의 예이다. 장애거리 $d_0(p,q)$를 계산하기 위해 초기에 $d_e(p,q)$범위안에 있는 장애물 o_1,o_2를 검색한 후 가시 그래프를 VG'를 생성한다. 객체 p에서 q까지의 장애거리는 최소 값을 갖는 경로인 $d_1(p,q)$가 되며, 장애거리 $d_1(p,q)$와 교차되는 장애물들이 검색대상이 된다. 이러한 과정은 장애거리와 겹치는 장애물들이 더 이상 존재하지 않을 때까지 반복 처리한다.



(그림 3) 장애거리 계산

```
Procedure Compute_Obstructed_Distance (VG',p,q)
/* VG':Visibility Graph, RTO:Obstacle R*-tree */
BEGIN
   dO(p,q)=Shortest_Path_Dist(VG',p,q);
   O'=set of vertices in VG';
   Onew=Euclidean_Range(RTO,q,dO(p,q));
   WHILE (O'⊂Onew)
   BEGIN
       FOR each obstacle o in Onew-O'
       Add_Obstacle(o,G');
       dO(p,q)=Shortest_Path_Dist(p,q,G');
       O'=Onew;
       Onew=Euclidean_Range(RTO,q,dO(p,q));
    END
END.
```

검색공간에 다수의 고정된 장애물이 광범위하게 존재하는 경우에는 장애 거리를 구하기 전에 장애물에 따른 정점들의 전체 쌍을 대상으로 가시 그래프를 생성해 놓음으로써 중복 연산을 피할 수 있다. 그러나 장애물의 수가 적거나 고정되어 있지 않는다면 연산시간

을 줄이기 위해 기준 반경 값 내에 놓이게 되는 정점들로만 대상을 제한하는 방법도 고려될 수 있다.

3. 장애물을 고려한 역최대근접질의 알고리즘

이 장에서는 장애물이 존재하는 검색공간에서 객체들의 위치속성이 고려된 검색거리 측도인 최소장애거리, 최적장애거리 그리고 최대장애거리를 소개하고 이를 이용한 역최대근접질의처리 알고리즘을 제안한다. 제안된 최소장애거리, 최적장애거리 그리고 최대장애거리들은 [1]에서 제시된 검색거리를 역최대근접질의의 비용을 최적화하기 위해 확장한 측도이다.

장애물이 존재하는 N 차원 검색공간에서 질의기준 Q와 최소경계사각형(minimum bounding rectangle: M)사이의 가장 가까운 거리인 최소장애거리 MINODIST는 다음과 같다.

$$MINODIST(Q,M) = \sum_{i=1}^{N} d_i^2(Q,M)$$

여기에서

$$d_i(Q,M) = |Q_i - M_i|$$

$$= \begin{cases} Q_i, & Q_{Li} > M_{Ui} \\ Q_{Ui}, & Q_{Ui} < M_{Li} \\ 0, & otherwise \end{cases}$$

$$M_i = \begin{cases} M_{Li}, & Q_{Ui} < M_{Li} \\ M_{Ui}, & Q_{Li} > M_{Ui} \\ 0, & otherwise. \end{cases}$$

(Q_{Li}, Q_{Ui} : 질의범위의 시작과 끝,
M_{Li}, M_{Ui} : 최소경계사각형의 시작과 끝).

정의된 MINODIST는 최소경계사각형 M에 포함되어 있는 부검색공간들 중에서 질의기준 Q에 가장 근접하고 있는 객체 또는 부검색공간을 결정하기 위한 거리이다.

색인의 검색 시 불필요한 노드의 검사를 피하고 다음 검색대상이 되는 노드의 수를 최소로 하는 검색거리 측도가 필요하다. 최적장애거리 OMINODIST는 N차원 검색공간에서 질의기준 Q에서 최소경계사각형 M을 구성하는 임의의 $N-1$차원을 포함할 수 있는 거리들 중 최소거리로 계산된다.

만약 Q와 M이 겹쳐있으면

$$OMINODIST(Q,M) = \min_i \{d_i(Q,M)\}$$

$$d_i(Q,M) = \min_{1 \le k \le N} \left\{ \begin{array}{l} (|qr_i - mr_i|^2 + \sum_{1 \le k \le N} |Qr_k - Mr_k|^2), \\ (|qr_i - Mr_i|^2 + \sum_{1 \le k \le N} |Qr_k - mr_k|^2), \\ (|q_i - Mr_i|^2 + \sum_{1 \le k \le N} |Q_k - mr_k|^2) \end{array} \right\}$$

여기에서

$$qr_i = \begin{cases} Q_i, & \text{if } \frac{(Q_{Li}+Q_{Ui})}{2} \le \frac{(M_{Li}+M_{Ui})}{2} \\ Q_{Ui}, & otherwise \end{cases}$$

$$Qr_k = \begin{cases} Q_{Lk}, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \le \frac{(M_{Lk}+M_{Uk})}{2} \\ Q_{Uk}, & otherwise \end{cases}$$

$$q_i = \begin{cases} Q_{Li}, & \text{if } \frac{(Q_{Li}+Q_{Ui})}{2} \le \frac{(M_{Li}+M_{Ui})}{2} \\ Q_{Ui}, & otherwise \end{cases}$$

$$Q_k = \begin{cases} Q_{Lk}, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \le \frac{(M_{Lk}+M_{Uk})}{2} \\ Q_{Uk}, & otherwise \end{cases}$$

$$mr_i = \begin{cases} M_{Li}, & \text{if } \frac{(Q_{Li}+Q_{Ui})}{2} \le \frac{(M_{Li}+M_{Ui})}{2} \\ M_{Ui}, & otherwise \end{cases}$$

$$Mr_k = \begin{cases} M_{Lk}, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \le \frac{(M_{Lk}+M_{Uk})}{2} \\ M_{Uk}, & otherwise. \end{cases}$$

그렇지 않으면

$$OMINODIST(Q,M) = \min\{d_i(Q,M)\}$$

$$d_i(Q,M) = \min_{1 \le i \le N} \left\{ \begin{array}{l} (|qr_i - mr_i|^2 + \sum_{1 \le k \le N} |Qr_k - Mr_k|^2), \\ (|qr_i - Mr_i|^2 + \sum_{1 \le k \le N} |Qr_k - mr_k|^2) \end{array} \right\}$$

여기에서

$$qr_i = \begin{cases} Q_{Li}, & Q_{Li} \le M_{Ui} \\ Q_{Ui}, & Q_{Ui} \le M_{Li} \\ Q_i, & otherwise \end{cases}$$

$$Q_i = \begin{cases} Q_{Li}, & \text{if } \frac{(Q_{Li}+Q_{Ui})}{2} \le \frac{(M_{Li}+M_{Ui})}{2} \\ Q_{Ui}, & otherwise \end{cases}$$

$$= \begin{cases} Q_k, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \quad \frac{(M_{Lk}+M_{Uk})}{2} \\ Q_{Uk}, & otherwise \end{cases}$$

$$mr_i = \begin{cases} M_{Li}, & \text{if } \frac{(Q_{Li}+Q_{Ui})}{2} \quad \frac{(M_{Li}+M_{Ui})}{2} \\ M_{Ui}, & otherwise \end{cases}$$

$$Mr_K = \begin{cases} M_{Lk}, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \quad \frac{(M_{Lk}+M_{Uk})}{2} \\ M_{Uk}, & otherwise \end{cases}$$

색인을 이용하여 주어진 질의 기준으로부터 가장 가까운 객체를 검색하는 동안 MINODIST와 OMINODIST의 차이가 아주 크거나 작은 경우가 발생할 수 있다. 이를 위해 질의기준과 최소경계사각형의 모든 꼭지점들 중에서 가장 먼 점까지의 거리인 최대거리 MAXODIST를 사용한다.

$$AXODIST\ Q,M) = \max\ d\ (Q,M)\}$$

$$(\ ,M) =$$

$$1 \quad \max_{i} \quad N \left\{ \begin{array}{l} (\mid qr_i - Mr_i \mid^2 + \quad i \quad k \quad \mid Qr_k - Mr_k \mid^2), \\ \quad 1 \quad k \quad N \\ \{(\mid q_i - Mr_i \mid^2 + \sum_{i} \quad k \quad \mid Q_k - Mr_k \mid^2) \\ \quad 1 \quad k \quad N \end{array} \right\}$$

여기에서

$$qr_i = \begin{cases} Q_{Li}, & \text{if } \frac{(Q_{Li}+Q_{Ui})}{2} \quad \frac{(M_{Li}+M_{Ui})}{2} \\ Q_{Ui}, & otherwise \end{cases}$$

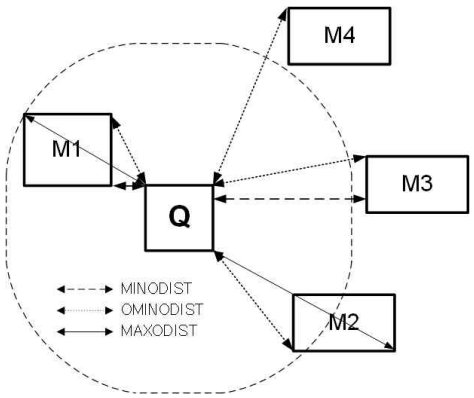
$$Qr_k = \begin{cases} Q_{Lk}, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \quad \frac{(M_{Lk}+M_{Uk})}{2} \\ Q_{Uk}, & otherwise \end{cases}$$

$$q_i = \begin{cases} Q_{Li}, & \text{if } \frac{(Q_{Li}+Q_{Ui})}{2} \quad \frac{(M_{Li}+M_{Ui})}{2} \\ Q_{Ui}, & otherwise \end{cases}$$

$$Q_k = \begin{cases} Q_{Lk}, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \quad \frac{(M_{Lk}+M_{Uk})}{2} \\ Q_{Uk}, & otherwise \end{cases}$$

$$Mr_k = \begin{cases} M_{Lk}, & \text{if } \frac{(Q_{Lk}+Q_{Uk})}{2} \quad \frac{(M_{Lk}+M_{Uk})}{2} \\ M_{Uk}, & otherwise. \end{cases}$$

MAXODIST를 사용함으로써 역최대근접질의 처리시 색인에서 방문할 노드의 수와 연산시간을 최소화할 수 있다.



(그림 4) 검색거리의 예

그림 4는 질의기준 Q로부터 MINODIST가 가장 작은 최소경계사각형 M1에 대한 MAXODIST를 기준으로 다음 방문대상에서 노드들을 제외하는 예이다. 역최대근접객체의 검색시 MAXODIST의 사용은 하나의 최소경계사각형에 포함되는 모든 객체 또는 부검색공간들과 비교가 가능하므로 색인에서 방문되는 노드의 수와 연산시간을 최소화할 수 있다.

역최대근접질의 알고리즘에서는 R*-트리[10]를 검색하는 동안 방문할 필요가 없는 노드들을 방문대상에서 제외하기 위해 OMINODIST 대신에 MAXODIST만을 이용하여 다음과 같은 전략을 사용한다.

ⅰ) 질의 기준 Q로부터 최소경계사각형 M′까지의 MAXODIST(Q,M′)보다 MINODIST(Q,M)가 더 큰 값을 갖는 최소경계사각형 M이 존재하면, M은 최대근접객체를 포함하지 않기 때문에 M에 해당하는 노드는 검색대상에서 제외한다.

ⅱ) 질의기준 Q로부터 객체 O의 거리가 최소경계사각형 M에 대한 MAXODIST(Q,M)보다 크다면, 객체 O를 최대근접객체 대상에서 제외한다.

ⅲ) 질의 기준 Q로부터 객체 O까지의 거리보다 더 큰 MINODIST(Q,M)를 갖는 모든 최소경계사

각형 M은 검색대상에서 제외한다.

본 논문에서 제안하는 역최대근접질의 알고리즘은 주어진 질의기준을 최대근접객체로 갖는 객체들을 찾기 위한 검색연산이 루트 노드부터 시작해서 하위 레벨의 노드까지 재귀적인 방법으로 이루어진다.

```
Procedure REVERSE_NN_SEARCH
BEGIN
  IF leaf_node(NODE) THEN
     sort_entry(Query Point,NODE.Entry);
     Prunning_Node_Entry(Query Point,
                         NODE.Entry);
     Get_Bucket(NODE.Entry,BUCKET);
     sort_entry(Query Point,BUCKET.Object);
     Prunning_Object(Query Point ,
                     BUCKET.Object);
     dist:=object_dist(Query Point,
                     BUCKET.Object);
     IF dist < REVERSE_NEAREST.dist
     THEN
       REVERSE_NEAREST.dist:=dist;
       REVERSE_NEAREST.object:=
                       BUCKET.Object;
     ENDIF
  ELSE
    sort_entry(Query Point,NODE.Entry);
    Prunning_Node_Entry(Query Point,
                        NODE.Entry);
    REVERSE_NN_SEARCH;
END.
```

알고리즘에서 sort_entry는 질의기준과 노드의 각 엔트리간의 MINODIST를 오름차순으로 순서화하는 함수이다. Prunning_Node_Entry는 MINODIST에 의해 순서화된 엔트리들을 MAXODIST와 비교 연산후 검색할 필요가 없는 노드들을 제거하는 절차이다. Prunning_Object는 버켓에 있는 각각의 객체들에 대해서 질의기준으로부터 객체까지의 거리보다 먼 MINODIST인 최소경계사각형들을 제거하는 절차이다. 또는 질의기준으로부터 객체까지의 거리가 최소경계사각형까지의 MINODIST보다 크다면, 객체를 최대근접객체 대상에서 제외하는 절차이다. 그리고 object_dist는 질의 기준과 버켓에 있는 객체들 사이의 거리를 계산하는 함수이다.

4. 결 론

다차원 검색공간에서 역최대근접질의 처리는 많은 디스크 접근과 질의처리시간을 요구한다. 또한 데이터의 차원이 증가함에 따라 검색비용이 크게 증가할 수 있다.

역최대근접질의의 처리비용을 최적화하기 위해서는 색인에서 검색되는 노드의 수를 최소화할 수 있는 검색거리 측도가 필요하다. 그리고 실세계에 존재하는 장애물들과 객체들의 위치속성을 고려해야 한다.

본 연구에서는 다차원 검색공간에서 장애물과 객체들의 위치속성을 고려한 검색거리측도인 최소장애거리와 최적장애거리 그리고 최대장애거리를 제시하고 이를 이용한 역최대근접질의의 처리 방법을 제안하였다.

제안된 방법은 최소장애거리, 최적장애거리와 최대장애거리를 조합하여 색인에서 방문대상이 되는 노드를 선정함으로써 장애물이 존재하는 검색공간에서 역최대근접질의에 따른 처리비용을 최소화할 수 있다. 그리고 암호화된 공간 데이터를 기반으로 하는 응용에서도 질의처리시 검색대상 및 범위를 줄이는데 매우 효율적이다.

앞으로의 연구과제는 본 논문에서 제안한 검색거리 측도들을 적용한 역최대근접질의 처리 알고리즘의 성능을 기존의 알고리즘과 다양한 응용 환경에서 비교 평가하는 것이다.

참고문헌

[1] 김홍기,“공간국부성을 최적화하는 클러스터링 방법,”정보과학회지:데이터베이스, 31권 2호, pp.83～90, 2004.

[2] 선휘준, “순환검색공간에서 K-최근접객체 쌍을 찾는 알고리즘에 관한 연구,” 한국공간정보학회지, 20권 2호, 2012.

[3] A.K.H.Tung, J.Hou, “Spatial Clustering in the Presence of Obstacles,” ICDE, pp.359～367, 2001.

[4] A.Singh, A.Tosun,“High Dimensional Reverse Nearest Neighbor Queries,”CIKM, pp.91～98, 2003.

[5] A.W.Yu, N.Mamoulis,“Reverse Top-k Search using Random Walk with Restart,”PVLDB, vol 7, no.5, pp.401～412, 2014.

[6] C.Yang, K.Lin,“An Index Structure for Efficient Reverse Nearest Neighbor Queries,” ICDE, pp.485～492, 2001.

[7] C.Xia, D.Hsu,“A Fast Filter for Obstructed Nearest Queries,” BNCOD, pp.203～215, 2004.

[8] F.Korn, S.Muthukrishnan,“Influence Sets based on Reverse Nearest Neighbor Queries,” SIGMOD, pp.201～212, 2000.

[9] I.Stanoi, D.Agrawai,“Reverse Nearest Neighbor Queries for Dynamic Databases,” SIGMOD Workshop DKMD, pp.44～53, 2000.

[10] N.Beckmann, H.Kriegel, R.Schneider and B. Seeger, “The R*-tree: an Efficient and Robust Access Method for Points and Rectangles,” Proc. ACM SIGMOD Int.Conf. on Management of Data, pp.322～331, 1990.

[11] R.Benetis, C.S.Jensen, “Nearest and Reverse Neighbor Neighbor Queries for Moving Objects,”VLDB Journal, vol.15, no.3, pp.229～250, 2006.

[12] R.Pereira, A.Agshikar, P.Keni,“Range Reverse Nearest Neighbor Queries,”Proc. of KICSS’2013, pp.509～518, 2013.

[13] X.Ma, C.Zhang, S.Shekhar, Y.Huang, H.Xiong,“On Multi-type Reverse Neighbor Search,” Data & Knowledge Engineering 70, pp.955～983, 2011.

[14] Y Gao, B Zheng, G Chen, WC Lee, KCK Lee, Q Li,“Visible Reverse k-Nearest Neighbor Queries,”Knowledge and Data Engineering, IEEE Transactions on 21 (9), 1314～1327, 2009.

[저자소개]

선 휘 준 (Hwi-joon Seon)
1989년 2월 전남대학교 계산통계학과 (이학석사)
1998년 2월 전남대학교 전산통계학과 (이학박사)
email : hjseon@sgu.ac.kr

김 홍 기 (Hong-ki Kim)
1984년 2월 전남대학교 계산통계학과 (이학사)
1986년 2월 전남대학교 계산통계학과 (이학석사)
1996년 2월 전남대학교 전산통계학과 (이학박사)
email : hkkim@dsu.ac.kr