다중선 최근접 객체 질의

정재화[O†] 장홍준[†] 정경호[†] 김성석[§] 길준민[§§] 정순영[†]
[†]고려대학교 컴퓨터교육과, [§]서경대학교 컴퓨터학과, [§§]대구카톨릭대학교 컴퓨터교육학과
bigbearian,white109,jungkh,jsy@comedu.korea.ac.kr, jmgil@cu.ac.kr, sskim03@skuniv.ac.kr

Polyline Nearest Neighbor Queries

JaeHwa Chung[O†], HongJun Jang[†], KyungHo Jung[†], SungSuk Kim[§], JoonMin Gil[§§], SoonYoung Jung[†]
[†]Dept. of Computer Science Education Korea University
[§]Dept. of Computer Science SeoKyeoung University
[§§]Dept. of Computer Science Education Catholic University of DaeGu

요 약

최근접 객체 질의(Nearest Neighbor Query)는 질의가 요청된 지점으로부터 가장 가까운 객체를 찾는 질의로 위치기반 서비스 분야에서 가장 널리 사용되고 있는 질의의 형태이다. 이를 기반으로 한 지역 최근접 객체 질의(Range Nearest Neighbor), 연속 최근접 객체 질의(Continuos Nearest Neighbor)등의 확장된 개념으로 다양한 최근접 객체 질의가 제안되어 왔다. 그러나 지금까지의 최근접 객체 질의를 기반으로 한 연구들은 점으로 표현된 질의를 기준으로 하여 최근접 객체를 찾는 기준점 최근접 객체(Point Nearest Neighbor) 질의를 기반으로 하고 있어, 점으로 표현이 불가능한 1차원 형태의 질의에 대하여 효과적인 최근접 객체를 검색하는 연구는 연구된 바 없다.

본 논문에서는 한 개 이상의 1차원 형태의 선분으로 이루어진 질의에 대하여 질의 주변의 객체 중 최근접 객체를 찾는 다중선 최근접 객체 질의(Polyline Nearest Neighbor)를 정의하고 효과적인 질의 처리 알고리즘을 제안하였다. 제안된 기법의 성능 분석을 위한 실험은 객체와 질의가 다양한 형태로 분포되어 있는 환경아래 진행되었으며, 실험 결과는 기대 값과 근접한 결과 값을 얻었다.

1. 서 론

사용자와 객체의 이동성이 증가하면서 다양한 종류의 공간 질의가 발생할 때 효율적인 질의 처리를 수행할 수 있도록 하는 많은 연구가 위치 기반 서비스(Location Based Services : LBS) 분야에서 진행되어 왔다. 최근접 객체 질의(Nearest Neighbor Query : NNQ)[1]는 실세계에서 사용되는 가장 대표적인 질의의 형태로 질의가 발행된 위치로부터 가장 가까운 객체를 찾는다. 최근접 객체 질의와 더불어 지역 객체 질의(Range Query : RQ)[2]는 질의가 지정한 지역에 포함된 객체를 결과 값으로 주는 질의로서 최근접 객체 질의와 함께 다양한 분야에서 활용되어왔다.

일반적으로 기준점 최근접 객체(Point Nearest Neighbor : PNN)로 불리는 질의는 사용자의 질의가 요청된 위치를 점(즉, 좌표축 상의 포인트)으[1]로 표현하고 그 점을 기준으로 공간상에서 가장 가까운 객체를 검색한다. 기준점 최근접 객체 질의의 확장된 형태로 질의 또는 객체가 이동하는 환경에서 지속적으로 최근접 객체를 유지하는 연속 최근접 객체 질의(Continuous Nearest Neighbor : CNN)[3]가 있다.

그러나 실세계에서 질의가 공간상에서 단 하나의 점으로 표현될 수 없는 경우가 존재한다. 예를 들어 그림 1에서와 같이 뉴욕을 방문 중인 관광객은 머물고 있는 호텔로부터 특정지역까지 정해진 이동경로에서 가까운 관광지(A~J)를 검색하고 싶을 것이다. 또한, 미시시피 강의 범람은 강에 가깝게 인접한 도시에 피해를 줄 것이며 재해방지를 위해서 강 주변 10Km 이내에 위치한 도시를 검색할 필요가 있다. 이와 같이 이동경로, 강, 산맥 등은 하나의 점으로 표현이 불가능한 질의 형태이다. 지금까지 연구에서 0차원의 한 점으로 표현될 수 없는 1차원 형태의 질의에 대하여 가장 가까운 객체를 효율적으로 검색하는 알고리즘 즉, 다중선 최근접 객체(Polyline Nearest Neighbor : PLNN) 질의를 처리하는 방법에 대한 연구는 이루어 지지 않았다.

그림 1 다중선 최근접 객체 질의의 예(구글맵 인용)

다양한 공간 검색 질의 연구에서는 대표적으로 사용되는 공간색인인 R-Tree[4] 및 유사 색인[5, 6]을 효과적으로 탐색하기 위하여 보조적으로 질의와 MBR 내부의 객체의 위치를 예측하기 위한 거리척도(metric)인 MINDIST(이하 *MID*)와 MINMAXDIST(이하 *MMD*)를 사용한다. 그러나 두 거리척도는 질의점과 MBR과의 관계만을 고려한 개념이므로 질의가 점이 아닌 선으로 표현되는 환경에서는 그대로 적용이 불가능하며 따라서 R-Tree의 효율적인 접근이 어렵다.

본 연구의 세부연구 내용을 요약하면 다음과 같다.

- 새로운 질의 형태인 PLNN 질의를 정의한다.
- PLNN 질의에 대하여 색인을 효율적으로 탐색하고 질의 처리 성능을 향상시킬 수 있는 새로운 거리척도

"이 논문 또는 저서는 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임" (KRF-2007-314-D00223)

를 제안한다.
- PLNN의 효과적인 처리 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 다중선 최근접 객체와 유사한 기존 질의를 살펴보고 3장에서 다중선 최근접 객체의 정의 및 가정 그리고 공간색인인 R-Tree를 효율적으로 이용할 수 있는 거리척도를 정의한다. 4장에서 새롭게 제안된 거리척도를 적용하여 최근접 객체를 검색하는 알고리즘에 대하여 기술하였고, 5장에서는 연구 결과에 대한 분석 내용을 설명하였다. 마지막으로 6장에서 결론을 제시하였다.

2. 관련 연구

2.1. 질의 종류

기준점 최근접 객체 질의의 처리 알고리즘은 질의를 하나의 점으로 표현하고 이 점으로부터 객체까지의 거리를 유클리드 거리(Euclidean distance)로 계산하여 질의 결과를 생산한다. PNN 질의는 크게 질의점으로부터 가장 가까운 객체를 찾는 최근접 객체(Nearest Neighbor : NN)와 질의점을 최근접 객체로 여기고 있는 객체를 찾는 역 최근접 객체(Reverse Nearest Neighbor : RNN)[7]로 구분된다. 지금까지 문헌에서 NN과 RNN은 다양한 환경에서 효율적으로 결과를 생산할 수 있는 알고리즘[5, 6]이 연구되었다. 특히, 무선기기의 증가로 질의 또는 객체가 이동하는 환경에서의 최근접 객체를 효과적으로 유지하는 알고리즘들이 연속 최근접 객체(Continuous Nearest Neighbor : CNN)[3] 등이 제안되었다.
한편 [8]은 질의가 1차원 이상 다차원의 형태로 일반화된 질의영역에서 최근접 객체를 찾는 지역 최근접 객체(Range Nearest Neighbor : RNN) 질의를 제안하였다. RNN[8]은 다차원 질의영역 주변에 다수의 객체가 존재할 때, 보로노이 다이어그램(voronoi diagram)[8]의 속성을 이용하여 다차원 질의영역 외곽선의 부분별 최근접 객체를 찾고 최종적으로 질의영역을 구성하는 모든 부분 외곽선 중 가장 작은 거리값을 갖는 객체를 결과로 반환한다. RNN[8]은 PLNN과 유사한 선 최근접 객체(LNN)에 대하여 정의하였으나 N차원으로 확장하기 위한 중간 개념이며 해결 방법이 본 연구에서 제안한 다중선(Polyline)에 대한 질의 처리가 불가능 하다.

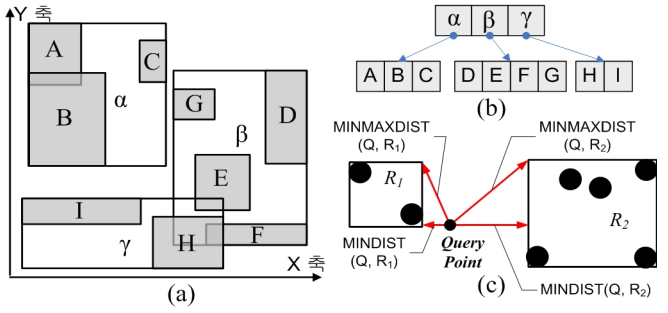


그림 2 공간색인 구조

2.2. 거리척도(metric)

공간색인에 대표적으로 사용되는 R-Tree 색인은 그림 2의 (a), (b)와 같이 공간을 최소경계 사각형(Minimum Bounding Rectangle : MBR)으로 한정하고 MBR을 계층적으로 연결하는 구조를 갖는다. MINDIST(이하 *MID*)와 MINMAXDIST(이하 *MMD*)는 R-Tree 및 유사 색인 구조에서 효과적인 탐색을 위해 제안되었다. MBR의 특성상 한 변에 반드시 하나 이상의 객체가 존재한다는 원리를 이용하여 *MID*와 *MMD*를 계산한다. *MID*는 MBR의 내부의 객체들과 질의간의 거리들 중 최소 거리를 의미한다. *MMD*는 MBR의 내부의 객체와 질의점 사이의 거리 중 반드시 하나 이상의 객체가 존재하는 거리들 중 최소 거리를 의미한다. 이 두 거리를 이용하여 그림 2 (c)에서 $MMD(Q,R_1)<MID(Q,R_2)$이므로 R_2에는 논리적으로 최근접 객체가 포함되지 않았음이 증명된다. 따라서 R_2는 검색과정에서 탐색할 필요가 없다.

3. 다중선 최근접 객체

다중선 최근접 객체(PLNN) 질의는 0차원의 형태의 질의에서 벗어나 무수한 점 질의가 나열되어 있는 1차원 형태의 질의에서 가장 가까운 객체를 찾는 질의를 의미한다. 0차원의 PNN 질의는 최근접 객체에 대한 검색영역이 하나의 원으로 생성되는 반면 다중선 질의는 검색 영역이 그림 3에서와 같이 민코스키 합(minkowsky sum)[9]의 형태로 형성되며 이 영역 내부에는 단 하나의 객체 즉 PLNN이 존재한다. 즉, 본 논문에서는 이 다중선 질의의 민코스키 합 영역을 효과적으로 검색하는 알고리즘을 제안한다. PLNN의 정의는 아래와 같다.

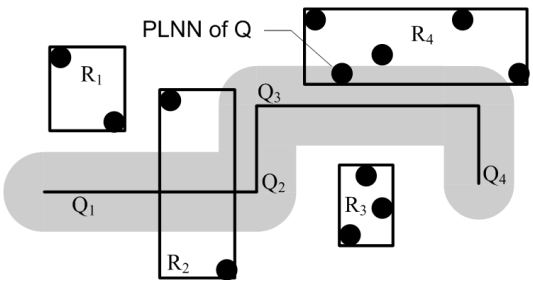


그림 3 다중선 질의의 최근접 객체 예

정의 (Polyline Nearest Neighbor(PLNN) Query) *주어진 객체 집합 S와 질의 Q에 대하여 다음 조건을 만족하는 객체 O_i를 찾는 질의를 다중선 최근접 객체 질의라고 한다. (1) $O_i \in S$ (2) $O_j \in S (1 \le j \le n)$인 모든 O_j에 대하여 $dist(Q_t,O_i)<dist(Q_u,O_j)$, 여기서 Q_t, Q_u는 O_i와 O_i에서 질의 Q상의 가장 가까운 한 점이며 dist()는 두 점 사이의 유클리드 거리를 나타낸다.*

3.1. 기본 가정 및 용어

공간상에서 객체는 0차원 한 점으로 표현되며 객체는 디스크 기반의 R-tree로 색인되어 있다. 또한 질의는 수직 또는 수평의 직선으로만 구성되며 질의를 구성하는 각각

의 직선은 질의선(query line)으로 지칭하였다. 질의 Q에 대하여 $Q^{h.s}$, $Q^{h.e}$은 각각 수평 질의선의 왼쪽 끝점, 오른쪽 끝점을 $Q^{v.s}$, $Q^{v.e}$는 수직 질의선의 아랫 끝점, 윗 끝점을 나타낸다. 마찬가지로 MBR에 대하여 R^l, R^r, R^b, R^t는 각각 R의 왼변, 오른변, 아랫변, 윗변을 R^{DL}, R^{UR}은 R의 좌하단, 우상단 꼭지점을 나타낸다. $\perp$은 수선을 의미한다. 따라서 $Q \perp R^l$은 R의 왼변에서 Q로 내린 연장선이 질의선 만나는 수선을 나타낸다.

3.2. PLNN의 새로운 거리

PNN에서 대표적으로 사용되는 *MID*와 *MMD*는 점으로 표현된 질의와 직사각형의 MBR 내에서의 객체 위치를 예측하며 효과적으로 색인을 사용하는 척도로서 이용된다. 그러나 그림 3과 같이 질의 Q가 4개의(Q_1 ~ Q_4)의 질의선으로 이루어져 있는 PLNN 질의인 경우 점과 직사각형 형태의 MBR과의 관계만을 고려한 기존의 두 거리척도는 최적화된 거리를 표현할 수 가 없다. 즉, 효과적인 색인 사용이 불가능하다. 예를 들면 질의선 Q_2에서 MBR R_3에 대한 Q_2 상에 위치한 임의의 많은 점간의 거리들 중 최소값이 거리척도로 결정된다.
기존의 점과 MBR과의 *MID*와 *MMD*를 재정의하여 선과 MBR과의 관계를 고려하였다. 이를 위하여 질의선과 MBR의 위치에 따라 질의선 상의 임의의 한 점에서 MBR까지의 *MID*과 *MMD* 값의 변화를 그래프로 도식화 하였다. 또한 질의선에서 최소의 거리척도의 값이 결정되는 위치에 따라 6개의 사례로 나누어 해결하는 상향식 접근(bottom-up approach) 방법을 적용한다. 그림 4는 질의선이 MBR의 외부에 위치하는 사례를 그림 5는 질의선이 MBR의 내부에 위치하는 사례를 표현하고 있다. 그래프에서 직선과 점선은 각각 *MID*와 *MMD*를 의미한다.

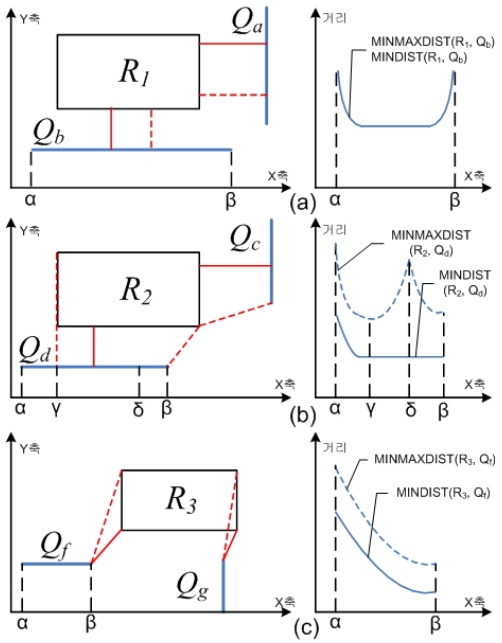


그림 4 사례 1, 2, 3

사례 1 질의선이 MBR의 외부에 존재하고 x축 또는 y축으로의 프로젝션 결과, MBR이 질의선에 포함되는 경우 : 그림 4 (a)
MBR R의 x축 또는 y축 상에서 존재하는 구간이 질의선 Q가 위치하는 구간 안에 포함되어 있는 형태로 그림 4(a)의 Q_a, Q_b와 같다. 그림 4(a)는 Q_b상의 임의의 점이 Q_b^s에서 Q_b^e까지 이동할 때 점의 이동 자취에 따른 $MID(Q_b, R_1)$와 $MMD(Q_b, R_1)$값의 변화를 나타내고 있다. Q_b에서 수평하고 가까운 R_1의 한 변을 R_1^h라고 할 때 *MID*는 $\gamma \le x \le \delta$ 구간에서 R_1^h에서 Q_b까지 내린 수선의 길이 $|Q_b \perp R_1^l|$로 결정된다. *MMD*의 경우 MBR은 구조적 특성상 임의의 한 변에 반드시 한 개 이상의 객체가 인접[1]하고 있는 특징을 이용한다. 따라서 $\gamma \le x \le \delta$ 구간에서 R_1^λ 상에 반드시 한 개 이상의 객체가 존재함이 보장되므로 사례 1에서의 *MMD*는 *MID*와 같은 값으로 결정된다.

$$MMD(Q, R) = MID(Q, R) \qquad (1)$$

그림 3에서 $MMD(Q_a, R_1)$과 $MMD(Q_b, R_1)$는 각각 $MID(Q_a, R_1)$, $MID(Q_b, R_1)$으로 결정됨을 알 수 있다.

사례 2 질의선이 MBR의 외부에 위치하고 x축 또는 y축으로의 프로젝션 결과, MBR과 질의선이 일부분만 겹쳐 있는 경우 : 그림 4 (b)
주어진 질의선과 MBR의 한 변이 평행하고 x 또는 y축 상으로 부분적으로 겹치는 경우 질의선의 한 끝점은 R의 외부에 다른 끝 점은 내부에 위치하게 된다. 그림 4 (b)의 오른쪽 그래프는 질의 Q_d상의 임의의 한 점에서 거리척도의 값을 변화를 보여주고 있다. Q_d에서 평행하고 가까운 R_2의 한 변을 R_2^h라고 하고 Q_d에 수직하고 가까운 변을 R_2^v 때, 그래프에서와 같이 $MID(Q_d, R_2)$는 $\gamma \le x \le \beta$ 구간에서 최소값을 유지하며 R_2^h상의 임의의 한 점으로부터 질의선까지 내린 수선의 길이 $|Q_d \perp R_2^h|$로 결정된다. *MMD*는 질의선의 양 끝점 Q_d^s, Q_d^e 중 x축 상에서 R의 위치구간 중간에 위치한 끝점을 Q_d^e라고 하면 Q_d^e에서 R_2^h 양 끝점 $R_2^{h.s}$, $R_2^{h.e}$까지의 거리 중 긴 쪽을 또는 질의선으로부터 평행한 다른 변으로부터 질의까지 내린 수선의 길이 중 짧은 거리를 *MMD*로 결정할 수 있다.

$$MMD = \min \begin{cases} \max(dist(Q_d, R_2^{h.s}), dist(Q_d, R_2^{h.e})) \\ dist(Q_d, R_2^h \perp Q_d) \end{cases} \qquad (2)$$

사례 3 질의선이 MBR의 외부에 존재하고 x축 또는 y축으로의 프로젝션 결과, 겹치지 않거나 또는 한 점으로 겹치는 경우 : 그림 4 (c)
주어진 질의선 양 끝점 중 MBR에 가까운 끝점에서 계산된 거리척도 값은 이 점을 제외한 질의선의 다른 점에서의 계산한 거리척도 값보다 짧다. 즉 MBR에 가까운 쪽의 끝점에서의 거리척도 값이 최소화된 점이다. 그림

4의 (c)에서의 오른쪽 그래프와 같이 질의선 Q_f상의 $x=\alpha$에서 계산된 $MID(Q_f, R_3)$과 $MMD(Q_f, R_3)$ 값은 β로 이동하면서 점점 감소하여 $x=\beta$에서 최소값을 갖는다. 따라서 사례 3에서의 MID와 MMD의 값은 질의선의 MBR에 가까운 쪽 끝점에서 결정된다.

사례 4 질의선의 한 끝점은 MBR의 외부에 다른 끝점은 MBR의 내부에 위치하는 경우 : 그림 5 (a)
질의선이 외부에서 MBR의 내부로 들어가는 형태를 나타낸다. Q_i가 통과한 변을 R_4^{v1}의 맞은편 변을 R_4^{v2}라고 할 때, 그림5의 (a)의 왼쪽 그래프에서 질의선 Q_i상의 임의의 한 점, $x=\gamma$에서 질의선이 R_4^{v1}을 통과하여 R_4의 내부로 진입하므로 MID는 $\gamma \le x \le \beta$ 구간에서 최소값 0으로 결정된다. 반면 MMD는 $x=\gamma$에서 최적화된 값을 갖는다. 그러나 질의선이 R_4^{v2}에 근접함에 따라 MMD는 최소값을 갖는다.

$$MMD(Q_i, R_4) = \min\begin{cases} MMD(Q_i, R_4^{v1}) \\ MMD(Q_i, R_4^{v2}) \end{cases} \quad (3)$$

그림 4의 (a)에서 주어진 질의선 Q_i와 R_4의 $MID(Q_i, R_4)$는 R_4 내부 위치한 임의의 Q_i상에서 0의 값을 가지며 $MMD(Q_i, R_4^{v1}) < MMD(Q_i, R_4^{v2})$이므로 $x=\beta$에서 최소값이 결정된다.

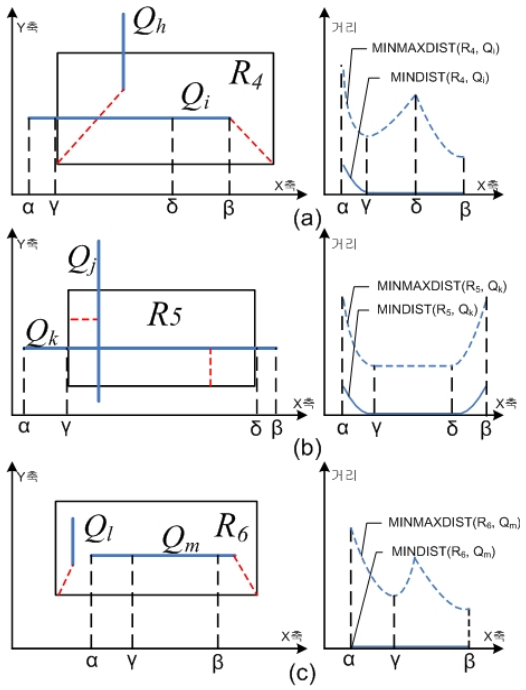


그림 5 사례 4, 5, 6

사례 5 질의선이 MBR의 양변을 모두 관통하여 양 끝점이 MBR의 외부에 위치한 경우 : 그림 5 (b)
질의선 Q가 MBR R을 관통할 경우 사례 1과 비슷하나 질의가 MBR의 내부에 위치함에 따라 질의선 Q상의 임의의 한 점에서의 거리척도 값의 변화는 그림 5의 (b)의 그래프와 같다. 질의가 x축 또는 y축 상으로 MBR의 내부에서 MBR을 포함하고 있는 형태이므로 $\gamma \le x \le \beta$에서 MID는 최소값 0을 갖고 MMD는 각각 $x=\gamma, \beta$ 일 때 최소값을 가지며 사례 1과 유사하게 질의선에 MBR에 평행하며 가까운 한 변 R_5^h까지 내린 수선의 길이로 정해진다.

$$MMD(Q_d, R_5) = dist(Q_d, R_5^h \perp Q_d), x = \gamma \text{ or } \beta \quad (4)$$

사례 6 질의선이 MBR의 내부에 완벽히 내포되어 양 끝점이 MBR의 내부에 위치한 경우 : 그림 5(c), 그림 6
질의선이 MBR의 내부에 위치하는 경우 MID는 질의선상의 모든 점의 자취에서도 값은 0으로 결정된다. 그러나 MMD는 그림 5(c)의 왼쪽 그래프와 같이 MBR의 크기에 대비하여 질의선의 길이와 질의선의 위치에 따라 결정되는 지점이 다양하다. 그림 6은 MBR을 각 변의 1/2 지점에서 나눠 사분면으로 분할하고 질의선의 위치, 길이에 따라 결정되는 MMD를 보여주고 있다. 주어진 질의 Q와 MBR R에서 질의선에 가깝고 평행한 변 R^h와 길이를 $|R^h|$라고 할 때, 그림 6의 질의 Q_1은 $|Q_1| \ge |R^h|/2$을 만족하고 한 개의 사분면에만 위치하는 경우를 나타내고 있다. MMD는 양 끝점에서 계산한 거리 중 작은값으로 결정된다.

$$MMD(Q_1, R) = \min\begin{cases} MMD(Q_1^s, R^h) \\ MMD(Q_1^e, R^h) \end{cases} \quad (5)$$

질의 Q_2는 $|Q_2| < |R^h|/2$을 만족하고 두 개의 사분면에 걸쳐있는 경우를 표현하고 있다. 질의선의 양 끝점은 1개의 사분면에 나뉘어 걸쳐져 있으므로 그림 6(b)와 같이 양 끝점 Q_2^s, Q_2^e에서 MMD값이 계산되는 MBR의 꼭지점 각각 υ, ν라고 할 때, MMD는 반대 꼭지점과의 거리로 결정된다.

$$MMD(Q_2, R) = \min\begin{cases} dist(Q_2^s, \nu) \\ dist(Q_2^e, \upsilon) \end{cases} \quad (6)$$

마지막으로 Q_3는 질의선이 $|Q_3| \ge |R^h|/2$을 만족하여 두 개의 사분면에 걸쳐있는 경우를 나타낸다. 질의선이 $|R^h|/2$보다 크면 MMD는 질의의 양 끝점에서 R^h의 양 끝점과의 가까운 쪽의 점과의 거리값 중 큰값으로 결정된다.

$$MMD(Q_3, R) = \max\begin{cases} dist(Q_3^s, R^{h \cdot s}) \\ dist(Q_3^e, R^{h \cdot e}) \end{cases} \quad (7)$$

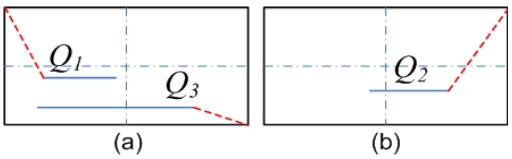


그림 6 질의선이 MBR 내부에 위치하는 경우

4. 다중선 최근접 객체 검색 알고리즘

지금까지의 연구에서 최근접 객체 검색 알고리즘은 깊이

우선 탐색(DFS)[1]과 너비우선탐색(BFS)[10]으로 나눌 수 있다. DFS 방식은 계층적으로 구성된 R-Tree 노드 중 근접 객체가 포함되어 있는 노드를 방문하고 재귀적으로 사전에 정의된 거리에 따라 하위 노드의 방문 순서를 결정한다. 관련연구 2.2에서 설명된 *MID*와 *MMD* 거리척도를 사용하여 효율을 높인다. 반면 BFS 방식은 하나의 우선순위 큐를 기반으로 하위 노드를 방문하며 매 방문 시 우선순위 큐를 *MID*값에 따라 정렬하고 방문 순서를 결정한다. 본 논문에서는 PLNN 질의를 R-Tree 색인과 깊이우선 탐색(DFS) 알고리즘 그대로 적용하였다. 그러나 색인 탐색의 효율을 높이기 위한 3가지 제거 휴리스틱(prune heuristic)를 적용하는 pruneBranchList 함수에서 기존에 사용되었던 MINDIST와 MINMAXDIST 함수를 대신하여 그림 7, 그림 8의 함수를 적용한다.

```
MINDIST(Q, R) 함수
입력 : 질의 Q 및 MBR R
for ( Q : Q_i )                    //질의를 질의선으로 분리
if (isIncluded(Q, R))
{
  // 사례 1, 2, 3
  if (Q의 끝 점이 R 완전 범위 포함=사례 1)
   then return getPointMin(R 범위 포함 점, r)
  if (Q의 끝 점이 R 부분 범위 포함=사례 2)
   then return getPointMin(R 범위 포함 점, r)
  if (Q의 끝 점이 R 범위 미 포함=사례 3)
   then return getPointMin(R 범위 가까운 점, r)
} else {                           // 사례 4, 5, 6
 return 0;
}
getPointMin 함수(ps, r)
Input ps : Points, r : MBR
Output ps으로부터 r의 mindist
```

그림 7 MINDIST 계산 알고리즘

본 논문에서는 DFS 알고리즘을 다중선 최근접 객체 검색에 그대로 적용할 수 있는 방안을 제시하고 있다. 따라서 k개의 근접 객체를 검색하는 PLkNN 질의로 확장하는데 아래와 같이 확장하여도 오류가 없다.

- k개의 최근접 객체를 저장할 수 있는 버퍼 확장
- k개의 최근접 객체 중 거리가 최대인 값을 기준으로 탐색 시 R-Tree 노드를 제거.

5. 연구 결과

4장에서 제안된 MINDIST, MINMAXDIST 알고리즘을 Java SDK 6.2과 NetBeans IDE 6.0.1을 사용하여 구현하고 다양한 객체 분포, 질의구성 형태에 따라 실험하였다. 실험은 AMD 3G 듀얼코어, 2G 메모리를 갖는 컴퓨터에서 수행되었다.

실험에 도입된 색인은 공간 질의에 대표적으로 이용되는 디스크 기반의 R*-Tree[11]를 사용하였다. 또한 전체 공간의 크기는 10000 X 10000으로 설정하고 분포형태의 종류에 따라 가상적 객체를 생성하였다. 본 연구에서 객체의 개수, 객체의 분포, 질의를 구성하는 질의선의 개수, 질의선의 길이 변화폭에 따라 질의 실행에 미치는 영향을 명확히 분석하기 위하여 다양하게 설정하였다. 표1은 실험에 사용된 파라미터 설정을 보여주고 있다. 기본 설정 값은 밑줄로 표현하였다.

```
MINMAXDIST(r, c) 함수
Input r : MBR, c : 사례 번호
Output minmaxdist
1.  if(사례 1,2,3)
2.  then points := genLinetoPoint(Line)
3.     return getPointMinmax(points, r)
4.  else if(사례 4,5)
6.  then points := genLinetoPoint(Line)
7.     return MINMAXDIST(r, 6);
8.  else              //사례 6
9.  return points와 r의 근접 모서리 길이값과
    points의 minmax값을 비교해서 작은값 선택
genLinetoPoint(l) 함수
Input l : Line
Output points
MBR과 Line의 위치에 따라 minmax를 구하는 Line위에
적합한 두 개의 점을 선택
getPointMinmax(ps, r) 함수
Input ps : Points, r : MBR
Output ps으로부터 r의 minmaxdist
```

그림 8 MINMAXDIST 계산 알고리즘

질의개수	5000 (균일분포)
객체개수	10K, 30K, <u>50K</u>, 70K, 90K, 110K, 130K, 150K
객체분포	<u>균일분포</u>, 지프분포
질의선 개수	5, 10, <u>20</u>, 30, 40, 50
질의길이	10, 30, <u>50</u>, 70, 90, 110, 130

표 1 파라미터 설정

실험에서 사용된 객체의 분포도는 균일분포와 지프분포[12]이다. 지프분포는 균일분포와 상반되는 형태의 비규칙적 분포 모델로서 x축 또는 y축의 값이 증가할수록 객체가 위치할 확률이 줄어 원점에 객체가 많이 몰려있는 패턴을 의미한다.

질의 처리 수행 능력은 질의 처리에 필요한 노드접근 수에 따라 평가된다. 따라서 각 실험 결과표는 질의처리에 필요한 접근한 노드의 수를 표시한다.

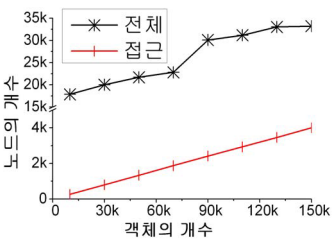


그림 9 객체수의 영향

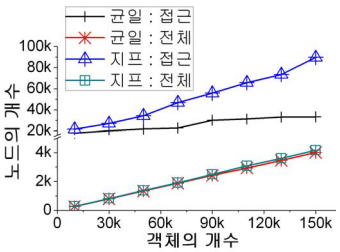


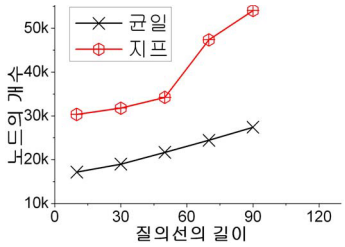
그림 10 객체분포의 영향

• **객체 수에 따른 결과 분석**

R-tree 색인에 삽입된 객체의 수에 따른 질의 처리 수행 능력을 평가하기 위하여 10K에서 150K까지 삽입하고 실험하였다. 그림 9에서 질의 처리에 접근한 노드의 개수는 선형적으로 증가하고 있다. 특히 R-Tree에서 질의 처리 능력의 급격한 저하를 유발하는 트리의 높이(height) 증가가 70K에서 90K사이에서 발생하였음에도 질의 처리에 필요한 노드의 수는 크게 영향 받지 않았음을 알 수 다.

• **객체 분포에 따른 결과 분석**

그림 10의 그래프 하단에서 균일/지프 분포 모두 객체 50K개에 대하여 비슷한 수의 노드가 R-Tree에 생성되었다. 그러나 질의처리에 접근된 수에는 확연한 차이가 보이는 것을 알 수 있다. 이는 객체의 개수가 증가할수록 증가하는 비율이 커지고 있다. 10K의 객체에 대하여 약 5%미만의 접근 노드수의 차이를 보이고 있으나 150K에 대하여 약 4배 가까운 차이를 나타낸다. 이는 지프분포는 객체가 원점 가까이에 집중적으로 위치하고 질의 처리 시 R-Tree 탐색과정에서 *MID*와 *MMD*로 제거되는 노드의 개수가 균일분포보다 지프분포가 적은것으로 분석된다.



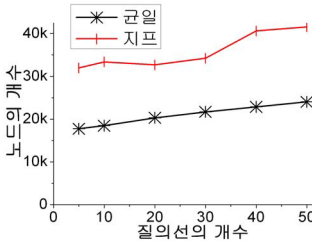


그림 11 질의선의 길이의 영향 그림 12 질의선 수의 영향

• **질의의 길이에 따른 결과 분석**

그림 11는 질의처리 수행에 질의선의 길이가 미치는 영향력을 보여주고 있다. 균일분포의 경우 질의선 길이의 변화에 따라 접근한 노드의 개수가 완만한 선형적인 증가를 보이고 있다. 반면에 불규칙 패턴인 지프분포의 경우 전체적으로 약 70% 이상 접근한 노드의 개수가 많고 또한 R-Tree의 높이가 증가하는 50K에서 70K구간에서 필요한 노드의 수가 급격하게 증가하였다. 이는 그림 9와 유사하게 지프분포에서 두 거리척도가 R-Tree 탐색의 효율을 높이지 못하는데 기인하는 것으로 분석된다.

• **질의선 수에 따른 결과 분석**

하나의 질의에 대하여 질의를 구성하는 질의선의 개수가 질의 처리에 미치는 영향을 그림 12에서 보여주고 있다. 타 실험과 유사하게 지프분포가 균일분포에 비하여 상대적으로 질의처리에 접근한 노드의 개수가 약 65% 증가하였다. 객체가 지프분포로 분포되어 있을 때 *MID*와 *MMD*로 R-Tree 탐색 시 제거 가능한 노드가 균일분포보다 감소한 것으로 분석된다. 그러나 그림 10의 지프분포 곡선과 같이 질의선의 길이가 증가함에 따라 균일분포와의 처리능력 차이가 커지는 것과는 달리 그림 11에서 질의의 질의선의 개수가 증가해도 균일분포와 일정 간격을 유지하며 증가하였다.

6. 결론

본 논문에서는 지금까지 최근접 객체 검색 연구에서 제안되지 않았던 공간상에서 선으로 표현되는 1차원 형태의 질의를 정의하고 효율적 검색 알고리즘의 필요성을 제안하였다. 또한 R-Tree 색인을 사용하여 효과적으로 최근접 객체를 검색하는 다중선 최근접 객체(PLNN) 질의 알고리즘을 제안하였다. PLNN 알고리즘은 한 개 이상의 선으로 이루어진 질의에 대하여 색인을 한번만 탐색하여 최근접 객체 결과를 생산하며 색인 탐색 과정에서 논리적으로 최근접 객체를 포함하지 않고 있는 MBR 방문을 최소화하기 위해 *MID*와 *MMD* 거리척도를 재정의 하였다. 질의점과 MBR을 고려한 기존 거리척도는 선형태의 질의에 대하여 최적화된 R-Tree 색인 탐색이 불가능하다. 즉 검색의 효율 저하를 피할 수 없다. 마지막으로 PLNN 알고리즘은 깊이 우선 탐색(DFS) 방법으로 결과를 생산하는 방법을 사용하였다. 실험 결과 객체가 균일하게 분포되어 있는 상황에서 질의처리에 필요한 접근 노드 개수가 선형적으로 증가한 반면 지프분포에서는 접근 노드 개수가 지수적의 비율로 증가하였다.

참고문헌

[1] Nick Roussopoulos 외 2명, Nearest Neighbor Queries, ACM SIGMOD, 71-79페이지, 1995년

[2] Dimitris Papadias 외 2명, Multi-Dimensional Range Query Processing with Spatial Relations, Geographical Systems, 343-365, 1997년

[3] Yufei Tao 외 2명, Continuous Nearest Neighbor Search, VLDB 2002년 8월

[4] Guttman, R-tree: A Dynamic Index Structure for Spatial Searching, ACM SIGMOD, 47-57페이지, 1984년

[5] Yannis Manolopoulos 외 3명, R-Trees : Theory and Applications, Springer-Verlag, 2006년

[6] Apostolos N. Papadopoulos 외 1명, Nearest Neighbor Search, Springer, 2005년

[7] Flip Korn 외 1명, Influence sets based on reverse nearest neighbor queries, ACM SIGMOD, 201-212페이지, 2000년 5월

[8] Haibo Hu 외 1명, Range Nearest Neighbor Query, TKDE 2006, 18권, 1호, 2006년 1월

[9] Joseph O'Rourke, Computational Geometry in C, Cambridge University Press, 1994년

[10] G.R. Hjaltason 외 1명, Distance Browsing in Spatial Databases, TODS, 24권, 2호, 265-318페이지, 1999년

[11] Norbert Beckmann 외 3명, R*-tree: An Effiecient and Robust Access Method for Points and Rectangles, ACM SIGMOD 1990, 19권, 2호, 1990년 6월

[12] http://en.wikipedia.org/wiki/Zipf's_law