

연속적인 스카이라인 질의를 위한 영역 결정 기법

나경석⁰ 김진호 박영배

명지대학교 컴퓨터공학과 데이터베이스연구실

{mjcomnks,solbong,parkyb}@mju.ac.kr

Region Decision for Continuous Skyline Queries

Kyoung-Seek Na⁰ Jin-Ho Kim Young-Bae Park

Dept. of Computer Engineering Myong Ji University

요약

최근에 이동 객체의 위치 정보를 활용한 위치기반서비스(Location-based Services : LBS)에 대한 관심이 급증하고 있고, 다양한 서비스들이 연구되고 있다. 기존의 이동 객체에 대한 위치 의존 질의(Location-dependent Query)들은 단순히 대상 객체와의 거리만을 고려하였고, 스카이라인 질의(Skyline Query)는 질의의 위치와 무관한 대상 객체의 정적인 속성만을 고려하였다. 이동 객체에 대한 스카이라인 질의는 스카이라인 질의의 다중 속성과 이동 객체의 동적인 속성인 대상 객체와의 거리를 고려해야 하기 때문에 이동 객체의 위치 변경에 따른 연속적인 질의가 발생한다. 이 논문에서는 이동 객체의 연속적인 스카이라인 질의를 효율적으로 처리하기 위한 Voronoi Diagram(VD)기반의 스카이라인 영역(Skyline Region)정의와 효율적인 영역 결정 기법을 제안한다.

1. 서론

최근 무선 인터넷 및 모바일 컴퓨팅 기술의 발전과 휴대폰, PDA, 노트북의 보급 확산과 더불어 위치확인기술(Location Detection Technology: LDT)을 이용해 이동 객체의 위치정보를 파악하고 주변 정보를 기준으로 다양한 정보를 제공하는 위치기반서비스(Location Based Service)를 개발하고 있다.

이동 객체는 시간에 따라 위치 정보가 계속 변경되는 공간 객체이며, 이동 객체의 현재 위치에 의존하는 질의(Location-dependent Query)가 발생 할 수 있다[1], 예를 들면 “현재 위치에서 가까운 식당을 검색하라”는 질의의 경우, 이동 객체가 질의 후 계속해서 위치를 변경하면 어떤 한 시점에서 계산된 결과는 다른 시간에서 유효하지 않을 수 있으므로, 위치 변경에 따라 연속적인 질의가 발생하며, 결과의 유효성을 검증하는 방법이 필요하다.

스카이라인 질의(Skyline Query)는 전체 객체 집합에서 객체의 여러 속성들을 다른 객체가 지배하지 않는 관심 있을만한 객체 집합을 검색한다[3]. “숙박료가 싸고, 해변과의 거리가 가까운 호텔 검색하라”의 예처럼 스카이라인 질의는 단일 속성이 아닌 다중 속성을 고려해야하고, 질의 결과는 단일 객체가 아닌 객체 집합이다. 이전 연구들에서 이동 객체에 대한 기법들[1][2]은 단순히 단일 속성인 대상 객체와의 거리만을 고려하였고, 스카이라인 질의에 대한 기법들[3][4][5][6]은 질의 위치와는 무관한 객체의 정적인 속성만을 고려하였다.

이동 객체에 대한 스카이라인 질의는 스카이라인 질의의 다중 속성과 이동 객체의 동적인 속성인 대상 객체와의 거리를 고려해야 한다. 예를 들어 해변 가로 놀러간 사용자는 “현재 위치에서 가장 값싸고 숙박료가 싸고, 해변과의 거리가 가까운 호텔을 검색하라”는 질의를 할 수 있다. 이 경우, 사용자는 이동 객체지만 단일 속성인 대상 객체와의 거리뿐만 아니라 다중 속성을 고려해야하고, 스카이라인의 결과는 대상 객체와의 거리에 따라 유효하지 않으므로 새로운 결과를 요청하기 위해 연속적인 질의가 필요하다.

이 논문에서는 이동 객체의 연속적인 스카이라인 질의를 효율적으로 처리하기 위해 이동 객체의 위치에 따라 스카이라인 질의 결과에 포함할 수 있는 대상 객체의 스카이라인 영역

(Skyline Region)을 정의하고, 영역을 결정하는 효율적인 방법을 제안한다.

2. 관련 연구

2.1 스카이라인 질의 (Skyline Queries)

스카이라인이란 전체 객체 집합에서 다른 객체가 지배하지 않는 관심 있을 만한 객체의 집합이다. 어떤 객체는 모든 차원 또는 적어도 하나의 차원이 좋을 경우 다른 객체를 지배한다 [1]. 그림 1은 해변과의 거리와 숙박료에 대한 호텔의 스카이라인을 나타낸다.

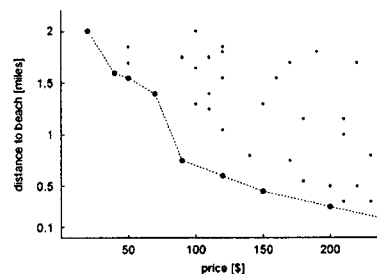


그림 1 호텔의 스카이라인

스카이라인 질의에 대한 알고리즘은 Block Nested Loop (BNL)과 Divide-and-Conquer(D&C)기법[3]이 제안 되었고, 비교 횟수를 줄이기 위해 Bitmap과 Index(B-tree)를 이용한 기법[4]이 제안 되었으며, 최근에는 R-tree기반의 NN(Nearest Neighbor) 검색[5]을 이용한 BBS (Branch-and-Bound Skyline) 기법[6]이 제안 되었다.

2.2 이동 객체를 위한 연속적인 최근접 질의 (Continuous Nearest Neighbor Query for Moving Objects)

이동 객체는 시간에 따라 위치 정보가 계속 변경되는 공간 객체를 의미한다. 이러한 이동 객체의 질의는 위치에 따라 결과는 유효하지 않을 수 있으므로 위치가 변경될 때마다 계속해서 새로운 결과를 요청해야 하기 때문에 위치 변경에 따라 연속적인 질의가 발생하며, 결과의 유효성을 검증하는 방법이 필요하다.[1][2]

연속적인 최근접 질의를 효율적으로 처리하기 위한 기법으로 R-tree에 정적인 객체 집합들의 Voronoi cell을 미리 계산하여 저장하는 기법[1]을 제안 했다. Voronoi cell이란 전체 정적 객체 집합 $P = \{p_1, p_2, \dots, p_n\}$ 에서 p_i 와 인접한 객체들의 수직 이동 분선들이 형성하는 영역이고, 이 영역 안에 있는 이동 객체의 최근접(Nearest Neighbor) 객체는 항상 p_i 다.

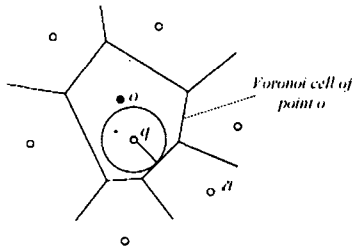


그림 2 Voronoi Diagram

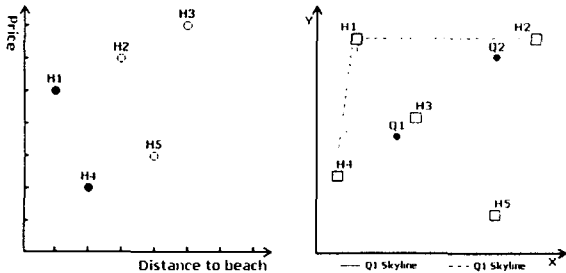
Voronoi cell을 이용하면 연속적인 질의에 대한 네트워크 부하를 줄이고, 결과의 유효성을 검증 할 수 있다.

3. 스카이라인 영역 결정

3.1 이동 객체에 대한 스카이라인 질의

객체의 정적 속성에 대한 스카이라인은 항상 고정이지만 이동 객체에 대한 스카이라인 질의는 위치 변경에 따른 대상 객체와의 거리라는 동적 속성 때문에 가변적이다.

이동 객체에 대한 스카이라인 질의 결과는 다른 객체들을 정적 속성과 동적 속성에 대해 지배하는 객체들로 이루어진다. 질의의 위치와는 상관없이 스카이라인 질의 결과는 정적 속성에 대해 지배하는 객체들을 항상 포함한다. 그러나 정적 속성에 대해 지배를 받을지라도 동적 속성 때문에 스카이라인 질의 결과에 포함될 수 있는 객체들도 존재한다. 즉 이동 객체에 대한 스카이라인 질의 결과는 객체의 정적 속성에 대해 지배하는 객체보다 이동 객체와 가까운 객체들도 포함한다.



(a) 호텔의 정적 속성 관계 (b) 질의 점 위치
그림 3 위치에 따른 스카이라인

예를 들어 호텔의 정적 속성 관계가 그림 3-(a)와 같은 경우 “현재 위치에서 가장 값이 싸고, 해변과의 거리가 가까

운 호텔을 검색하라.”는 질의의 결과는 객체의 정적 속성에 대한 스카이라인 H1과 H4를 위치에 관계없이 포함하고, Q1은 동적 속성에 의한 H3, Q2는 H2를 포함한다.

3.2 스카이라인 영역

이동 객체가 위치를 변경할 때마다 질의 위치와 전체 객체의 거리를 계산하여 새로운 스카이라인을 재계산하는 것은 비효율적이다. 이동 객체에 대한 질의는 대상 객체의 스카이라인 영역을 미리 지정하고, 현재 위치와 영역간의 관계를 판별함으로써 위치를 변경할 때마다 전체 객체에 대해 재계산하지 않고 새로운 스카이라인을 계산할 수 있다.

스카이라인 영역(SR: Skyline Region)이란 전체 대상 객체 집합 D의 각 원소 H_i 를 이동 객체가 스카이라인으로 포함하는 영역이다. 즉 H_i 를 정적 속성으로 지배하는 객체들의 집합 SD의 각 원소 H_j 와 H_i 에 의해 생성되는 수직 이동분선들이 형성하는 영역이다. SR 영역에 위치한 이동 객체에 대한 스카이라인 질의 결과는 SD의 각 원소보다 상대적으로 가깝기 때문에 H_i 를 포함한다.

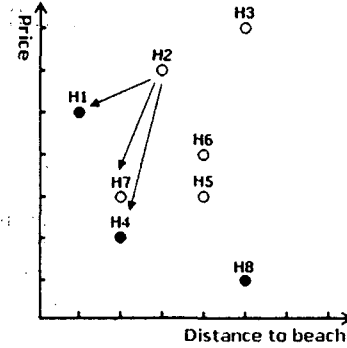


그림 4 호텔의 정적 속성 관계

그림 4와 같이 객체 집합 D의 원소인 호텔 H2를 객체의 정적 속성으로 지배하는 호텔들의 집합 SD는 H1, H4, H7이며, H2와 SD의 각 원소 H1, H4, H7과의 수직 이동분선들이 형성하는 스카이라인 영역은 그림 5와 같다. 이 영역 안에 이동 객체가 위치하는 경우 H2는 SD의 각 원소보다 상대적으로 가깝기 때문에 스카이라인에 포함된다.

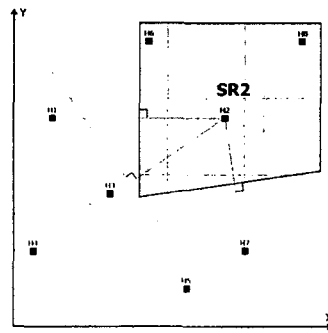


그림 5 H2의 스카이라인 영역

그림 6은 모든 스카이라인 영역을 나타내고 있으며, Q1을 포함한 영역은 SR3, SR6이므로 Q1의 스카이라인은 정적 속성에 의한 H1, H4, H8과 동적 속성에 의한 H3, H6이다.

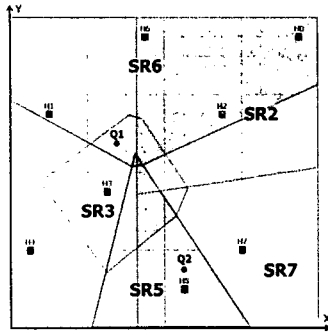


그림 6 전체 스카이라인 영역

3.3 스카이라인 영역 표현

좌표 평면 위의 어떤 도형에 의하여 만들어지는 영역은 부등식으로 나타낼 수 있다. 직선의 방정식을 $ax + by + c = 0$ 이라 할 때, 직선의 위부분 영역의 부등식은 $by > ax + c$ 이고 아랫부분 영역은 $by < ax + c$ 이다.

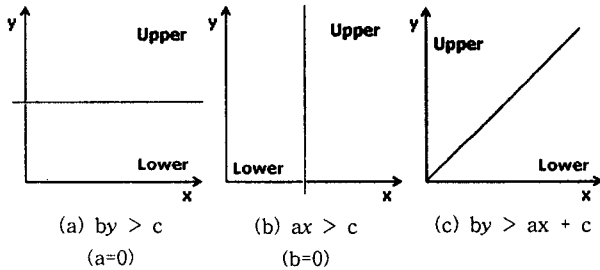


그림 7 부등식 영역

대상 객체의 위치에 따라 수직이동분선을 부등식으로 표현하고, 연립부등식을 이용하여 각 대상 객체의 스카이라인 영역을 표현한다.

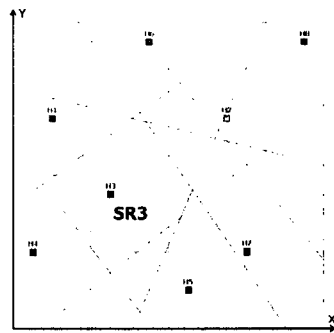


그림 8 H3의 스카이라인 영역

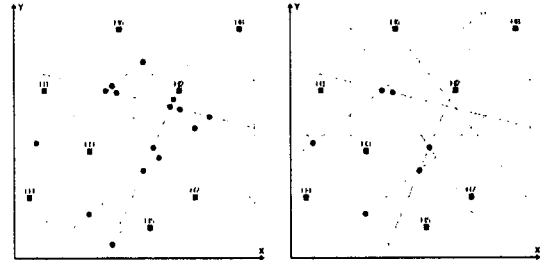
그림 8은 H3의 연립부등식의 영역을 나타내며 이동 객체의 위치에 따라 연립부등식의 조건을 검사하여 H3의 스카이라인 영역에 포함되는지 판별할 수 있다.

3.4 최적화된 스카이라인 영역 결정

그림 8에서 H3과 H8의 수직이동분선은 스카이라인 영역을 결정하지 않는다. 이러한 선분을 제거하여 최적화된 영역을

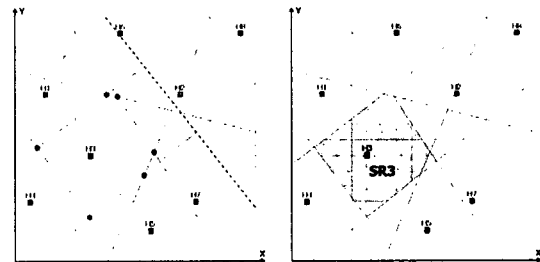
형성하는 선분을 결정함으로써 불필요한 비교 횟수를 줄일 수 있다. 최적화 영역을 결정하기 위한 3단계는 다음과 같다.

- 1단계 : 수직이동분선들의 모든 교차점을 계산.
- 2단계 : 연립부등식을 만족하는 교차점을 계산(SR_i 꼭지점 계산)
- 3단계 : 꼭지점 2개를 지나지 않는 선분 제거



(a) 1단계 : 교차점 계산

(b) 2단계 : 꼭지점 계산



(c) 3단계 : 선분 제거

(d) 최적화된 영역

그림 9 최적화 영역 결정 과정

4. 결론

이동 객체에 대한 스카이라인 질의는 다양한 위치 기반 서비스를 제공할 수 있다. 이 논문에서는 모든 대상 객체에 대해서 스카이라인에 포함될 수 있는 최적화된 스카이라인 영역을 미리 계산하여 이동 객체에 대한 스카이라인 질의를 효율적으로 처리하는 기법을 제안 하였다.

참고문헌

- [1] Zheng B., Lee, D. "Semantic Caching in Location- Dependent Query Processing" SSTD, p.97-116, 2001
- [2] J. Zhang, M. Zhu, D. Papadias, Y. Tao, and D. L. Lee. Location-based spatial queries. In SIGMOD, p.443-454, 2003
- [3] Borzsonyi, S, Kossmann, D., Stocker, K. "The Skyline Operator" In ICDE, p.421-430, 2001.
- [4] Tan, K., Eng, P. Ooi, B. "Efficient Progressive Skyline Computation" In VLDB, p.301-310, 2001.
- [5] D. Kossmann, F. Ramsak, S. Rost, "Shooting Stars in the Sky: an Online Algorithm for Skyline Queries." In VLDB, p.275-286, 2002
- [6] Papadias, D., Tao, Y., Fu, G., Seeger, B. "An Optimal and Progressive Algorithm for Skyline Queries." In SIGMOD, p.443-454, 2003.