

# 위치기반 서비스를 위한 거리공간에서의 효율적인 연속 리버스 스카이라인 질의 처리 기법

(An Efficient Continuous Reverse Skyline Query Processing  
Method in Metric Spaces for Location-based Services)

임 종 태 <sup>\*</sup> 박 용 훈 <sup>\*</sup> 서 동 민 <sup>\*\*</sup> 유 재 수 <sup>\*\*\*</sup>  
(Jongtae Lim) (Yonghun Park) (Dongmin Seo) (Jaesoo Yoo)

**요약** 스카이라인을 응용한 다양한 질의처리기법 중에 하나로 리버스 스카이라인이 연구되었다. 하지만 리버스 스카이라인을 이용한 다양한 위치기반서비스를 제공하기 위해서는 지금까지 연구된 리버스 다른 스카이라인만으로는 제공할 수 있는 서비스에 한계가 있으며, 질의처리를 위한 비용도 크다는 단점이 존재했다. 본 논문에서는 거리공간의 객체정보를 대상으로 질의처리가 가능한 효과적인 연속 리버스 스카이라인 처리 기법을 제안한다. 제안하는 기법의 우수성을 보이기 위해서 제안하는 기법과 기존에 연구된 리버스 스카이라인 질의처리기법에 대한 성능평가를 수행하고 비교 분석했다. 그 결과, 기존의 기법보다 제안하는 기법이 연속질의를 처리하는데 우수한 성능을 보이는 것을 확인했다.

**키워드 :** 리버스 스카이라인, 거리공간, 위치 기반 서비스

**Abstract** As the variant of skyline query processing, reverse skyline queries have been studied. However, the existing methods for processing reverse skyline queries have the limitation of service domains and spend high costs to provide various location-based services. In this paper, we propose a new reverse skyline query processing method that efficiently processes a query with the objects in metric spaces. In addition, the proposed method also processes continuous reverse skyline queries efficiently. In order to show the superiority of the proposed scheme, we compare it with the previous reverse skyline query processing scheme in various environments. As a result, the proposed method achieves better performance than the existing method.

**Key words :** Reverse skyline query processing, Metric spaces, Location-based services

---

· 본 연구는 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업과 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업의 결과임(No. 2009-0089128)

\* 학생회원 : 충북대학교 정보통신공학과  
jtlim@chungbuk.ac.kr  
yhpark1119@chungbuk.ac.kr

\*\* 정회원 : 한국과학기술정보연구원 정보기술연구실 연구원  
dmseo@kisti.re.kr

\*\*\* 종신회원 : 충북대학교 정보통신공학과 교수  
jis@chungbuk.ac.kr  
(Corresponding author)

논문접수 : 2010년 8월 20일  
심사완료 : 2010년 9월 14일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용 행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 데이터베이스 제37권 제5호(2010.10)

## 1. 서 론

최근 넷북, 스마트폰 같은 모바일 기기의 보급이 확산되고 모바일 기기에서 서비스 가능한 많은 응용 프로그램들이 나오고 있다. 응용 프로그램들 중에 가장 큰 비중을 차지하는 것은 위치를 확인하고 주변정보를 이용하여 유용한 정보를 제공하는 위치기반서비스(Location-Based Service: LBS)이며, 이를 위한 많은 이동객체 질의처리기법이 연구되었다. 이동객체 질의처리기법 중에서도 두 개 이상의 다중 속성을 동시에 고려할 수 있는 스카이라인 질의처리기법(Skyline Query)이 활발하게 연구되고 있다[1-3]. 스카이라인은 적어도 하나 이상의 속성이 다른 객체들보다 나쁘지 않아서 질의가 고려해야 할 객체들을 검색해주는 질의처리기법이다. 스카이라인과 스카이라인을 응용한 다양한 처리기법이 연구됨

에 따라서 스카이라인을 이용한 다양한 위치기반서비스를 효과적으로 서비스 할 수 있게 되었다.

위치기반서비스를 제공하기 위해 스카이라인을 용용한 다양한 처리기법 중에 하나로 리버스 스카이라인인 연구되었다[4-7]. 리버스 스카이라인은 질의를 내리는 객체를 스카이라인으로 여기는 객체들을 검색하는 질의 처리기법이다. 스카이라인을 이용하여 '현재 위치로부터 가깝고 가격이 저렴한 상점'과 같은 질의를 처리할 수 있다면, 리버스 스카이라인을 이용하면 '우리 상점을 가격이 저렴하고, 거리를 가깝게 여기는 고객'과 같은 질의를 처리할 수 있게 되었다. 이런 리버스 스카이라인을 이용하여 과거보다 유용한 서비스를 제공할 수 있게 되었다.

기존에 연구된 대부분의 리버스 스카이라인은 다이나믹 스카이라인(Dynamic Skyline)을 기반으로 한 리버스 다이나믹 스카이라인(Reverse Dynamic Skyline)이다[4-7]. 하지만 다이나믹 스카이라인은 일반적인 스카이라인과 달리 유사성 질의를 처리하는 특수한 스카이라인이다. 따라서 리버스 스카이라인을 이용한 다양한 위치기반서비스를 제공하기 위해서는 기존에 연구된 리버스 다이나믹 스카이라인만으로는 제공할 수 있는 서비스에 한계가 있다. 또한 기존 기법은 속성값을 축으로 가지는 논리공간에서 질의를 처리하므로 좌표를 통해 거리를 일일이 계산하여 질의처리하기 때문에 비용이 크다는 단점이 존재했다.

따라서 최근 논문에서는 일반적인 스카이라인에 적용 가능하면서도, 객체들의 실제 좌표를 x축, y축 상에 표현한 거리공간에서 직접 질의 처리하여 위치기반서비스에 적용 가능한 리버스 스카이라인 처리기법을 제안했다[8]. 제안한 리버스 스카이라인 기법은 스카이라인에서 객체지배관계와 수직이동분선의 성질을 이용하여 거리공간에서 효율적으로 리버스 스카이라인 질의를 처리한다. 하지만 소개하는 기법을 이용하여 연속적인 질의를 처리하기 위해서는 최종검사 단계에서 매우 많은 영역질의가 필요하기 때문에 많은 질의처리 비용이 소모된다. 따라서 본 논문에서는 최종검사 단계에서 발생하는 영역질의를 줄일 수 있는 새로운 리버스 스카이라인 질의 처리 기법을 제안한다. 제안하는 기법의 우수성을 보이기 위해서 제안하는 기법과 기존에 연구된 리버스 스카이라인 질의처리기법의 성능평가를 수행하고 비교 분석했다. 그 결과, 기존의 기법보다 제안하는 기법이 연속질의를 처리하는데 우수한 성능을 보이는 것을 확인할 수 있었다.

본 논문의 구성을 아래와 같다. 2장에서는 스카이라인과 지금까지의 리버스 스카이라인에 대한 연구들을 소개한다. 3장에서는 본 논문에서 제안하는 거리공간에서

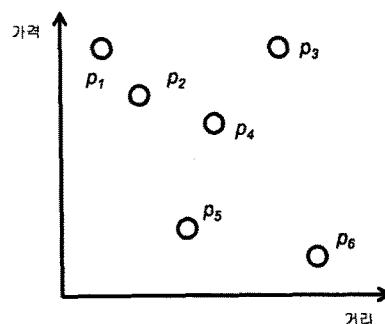
효율적으로 리버스 스카이라인 연속질의를 처리하는 기법을 소개한다. 4장에서는 제안하는 기법과 기존에 연구된 리버스 스카이라인 질의처리기법의 성능평가를 수행하여 비교 분석하고 5장에서 결론과 향후 연구를 기술하고 본 논문의 기술을 마친다.

## 2. 관련연구

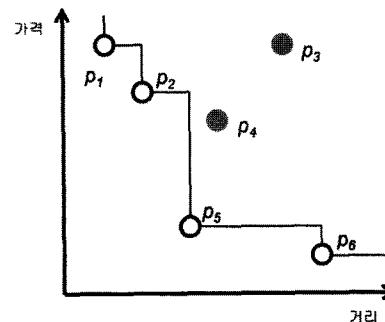
### 2.1 스카이라인(Skyline Query)

스카이라인이란 다중 속성을 고려하여 질의의 요구에 적합한 객체를 검색하는 질의처리기법이다. 스카이라인은 객체의 속성이 다른 객체보다 나쁘지 않아서 질의가 관심을 가질만한 객체를 질의의 요구에 적합한 결과로 가진다. 예를 들어 고객이 가격이 저렴하고, 현재 위치로부터 거리가 가까운 음식점을 검색한다면 질의는 고객이 되고, 객체는 음식점들이 된다. 그리고 각각의 객체는 가격과 질의로부터의 거리라는 속성 값을 가진다.

그림 1은 기본적인 스카이라인의 처리과정을 보여준다. 그림 1(a)는 스카이라인 처리를 위해 객체를 가격과 거리로 이루어진 논리공간상의 좌표에 위치시킨 그림이다. 그림과 같이 위치되고 나면 객체들 중에 다른 객체에 의해서 지배당하는 객체를 걸러내는 과정을 수행한



(a) 논리적인 공간상에 객체



(b) 스카이라인 질의 결과

그림 1 스카이라인 질의 처리 예제

다. 지배당하는 객체란 속성이 다른 객체보다 좋지 않아서 질의의 관심 대상이 되지 못하는 객체를 말한다.  $p_3$ 와  $p_4$ 의 경우  $p_2$ 나  $p_5$ 보다 가격도 비싸고 질의로부터 거리도 멀기 때문에 질의는  $p_3$ ,  $p_4$ 보다는  $p_2$ ,  $p_5$ 를 관심 있는 객체로 여길 것이다. 이때  $p_2$ ,  $p_5$ 는 지배하는 객체라고 하며 스카이라인 결과에 포함된다.  $p_3$ 나  $p_4$ 는 지배당하는 객체라고 하며 스카이라인 결과에 포함되지 않는다. 그리고  $p_1$ ,  $p_6$ 의 경우는 지배를 하고 있지는 않지만 지배를 당하지도 않기 때문에 결과에 포함된다. 이렇게 스카이라인을 수행하고 나면 지배를 당하고 있는  $p_3$ ,  $p_4$ 를 제외한 그림 1(b)와 같은 결과를 얻는다.

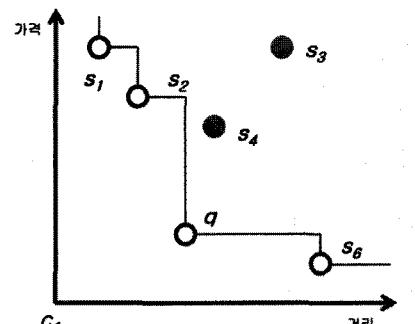
## 2.2 리버스 스카이라인(Reverse Skyline Query)

리버스 스카이라인은 질의를 내리는 객체를 스카이라인으로 여기는 객체들을 검색하는 질의처리기법이다. 리버스 스카이라인의 연구됨에 따라서 '우리 상점을 가격이 저렴하고, 거리를 가깝게 여기는 고객'과 같은 질의를 처리할 수 있게 되었다.

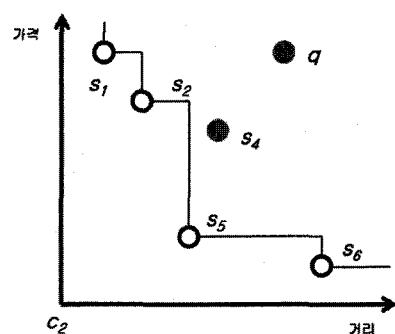
그림 2는 리버스 스카이라인 대상 객체들의 스카이라인을 보여준다. 그림 2(a)는 객체  $c_1$ 의 스카이라인을 보여주는 그림이다.  $c_1$ 의 스카이라인처럼  $q$ 가 스카이라인 결과에 포함되어 있다면,  $q$ 의 리버스 스카이라인 결과에는  $c_1$ 이 포함된다. 그림 2(b)는 객체  $c_2$ 의 스카이라인을 보여주는 그림이다.  $c_1$ 의 스카이라인과 반대로  $c_2$ 의 스카이라인처럼  $q$ 가 스카이라인 결과에 포함되지 않고, 다른 객체에게 지배를 당한다면  $q$ 의 리버스 스카이라인 결과에는  $c_2$ 가 포함되지 않는다. 리버스 스카이라인의 가장 기본적인 처리 기법은 GRSL(Greedy Reverse Skyline)[4]이다. 이 기법은 스카이라인의 지배관계를 이용하여 모든 객체에 대해서 스카이라인을 구하고 질의의 주체가 되는 객체가 객체들의 스카이라인 결과에 포함되어 있는지 확인하는 기법이다. 하지만 모든 객체에 대해서 스카이라인을 수행하여야 하기 때문에 처리 시간과 비용이 상당히 많이 든다는 단점이 존재한다. 최근 리버스 스카이라인을 효과적으로 처리하기 위해 많은 기법이 연구되었다. 스카이라인의 BBS(Branch and Bound Skyline)와 유사한 기법을 사용한 BPRS(Branch and Bound Processing of Reverse Skyline Queries)와 BPRS의 영역질의 횟수를 줄이기 위한 RSSA(Reverse Skyline Computation Using Skyline Approximations)가 처음으로 소개되었으며[4], 또한 일정한 범위를 가진 확실하지 않는 데이터에 대한 리버스 스카이라인 처리 기법도 제안되었다[6]. 또한 슬라이딩 윈도우 기법을 이용하여 리버스 스카이라인의 스트림 데이터를 처리하는 기법도 제안되었다[7].

## 2.3 거리공간에서의 리버스 스카이라인 질의 처리

기존에 연구된 리버스 스카이라인은 다이나믹 스카이



(a) 리버스 스카이라인 결과인  $c_1$ 의 스카이라인



(b) 리버스 스카이라인 결과가 아닌  $c_2$ 의 스카이라인

그림 2 리버스 스카이라인 대상 객체들의 스카이라인 비교

라인(Dynamic Skyline)을 기반으로 한 리버스 다이나믹 스카이라인(Reverse Dynamic Skyline)이다[4-7]. 하지만 리버스 다이나믹 스카이라인 처리기법은 거리의 차이를 이용하기 때문에 방향성이 존재하지 않아 일반적인 리버스 스카이라인 처리에 사용할 수 없으며, 객체 개수만큼의 거리계산 비용이 소모된다. 따라서 리버스 스카이라인을 이용한 다양한 위치기반서비스를 제공하기 위해서는 지금까지 연구된 리버스 다이나믹 스카이라인만으로는 제공할 수 있는 서비스에 한계가 있으며, 질의 처리를 위한 비용도 크다는 문제가 존재한다. 그래서 최근 거리공간에서 방향성을 고려한 리버스 스카이라인 처리기법이 제안되었다[8]. 이 기법은 스카이라인에서의 객체의 지배관계와 이동분선의 성질을 이용하여 거리공간에서 효율적으로 리버스 스카이라인 질의를 처리한다. 질의로부터 가까운 객체를 검색하여 거리가 아닌 다른 속성으로 질의를 지배할 수 있는 참조객체들을 찾고, 이동분선을 그려 질의와 참조객체 사이의 거리관계를 이용하면, 스카이라인을 그렸을 때 참조객체에 의해 질의가 지배당하는 객체들을 찾아 필터링 한다. 따라서 질의 근처에 위치하는 최소한의 객체들만으로 리버스 스카이라인 질의를 처리한다.

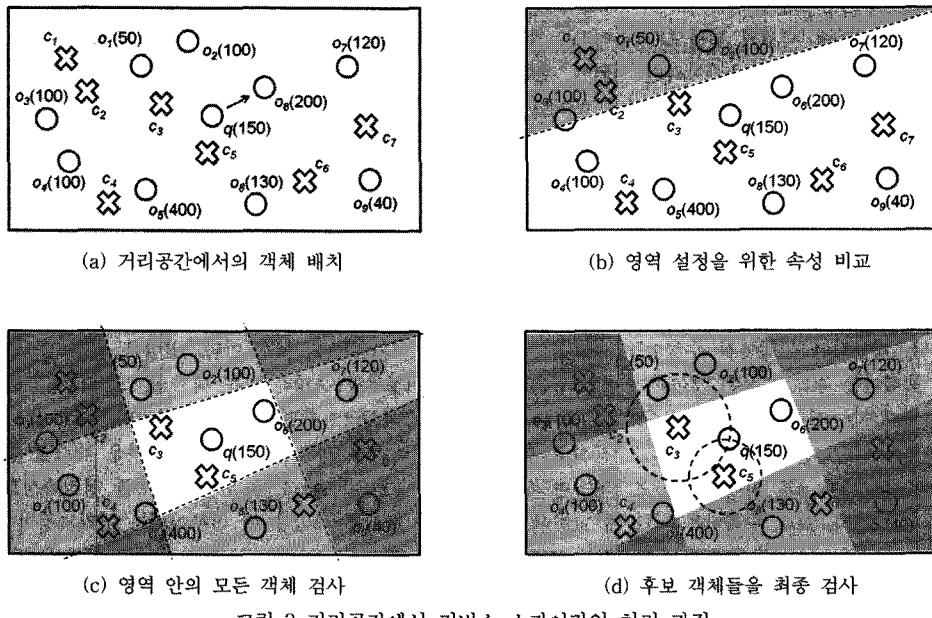


그림 3 거리공간에서 리버스 스카이라인 처리 과정

그림 3은 거리공간에서의 리버스 스카이라인 질의 처리과정을 나타낸 그림이다. 먼저 그림 3(a)는 위치속성에 의해 객체들이 거리공간에 위치한 그림으로 팔호 안에 있는 수치는 객체의 정적속성이다. 그림과 같이 위치하고 나면 질의의 주체가 되는 객체  $q$ 는 가까운 객체 중에  $o_1$ ,  $o_2$  등과 같이  $q$ 와 같은 타입의 객체들로부터 거리가 아닌 속성으로  $q$ 를 지배하는 참조객체를 검색한다. 그림 3(a)와 같이 최초로  $o_6$ 을 선택하고 정적속성들을 비교한 결과  $q$ 가 지배를 당하지 않으면  $o_6$ 은 다른 객체들이  $q$ 를 스카이라인으로 여기는데 영향을 주지 못하는 객체이므로 고려대상에서 제외된다. 그림 3(b)와 같이 다음 가까운 객체인  $o_2$ 와 비교한다. 정적속성들을 비교한 결과  $q$ 를 지배하는 참조객체라면 두 객체의 최단거리를 수직 이등분하는 선을 긋는다.  $q$ 를 기준으로 이등분선 바깥쪽에 위치하는 객체  $c_1$ ,  $c_2$ 의 스카이라인에서  $q$ 는 거리, 정적속성 모두  $o_2$ 의 지배를 받기 때문에  $c_1$ ,  $c_2$ 는 리버스 스카이라인의 결과에 포함될 수 없다. 이런 방법으로 가까운 객체부터 차례로 검색해 나가면 그림 3(c)와 같이 영역을 설정할 수 있다. 영역이 설정되면  $c_3$ ,  $c_5$ 와 같은 리버스 스카이라인 후보객체가 결정된다. 마지막으로 그림 3(d)와 같이  $q$ 보다 더 가까운 위치에  $q$ 를 지배하는 다른 객체가 있는지 확인하는 검사를 마치면  $q$ 에 대한 리버스 스카이라인의 처리가 모두 끝나게 된다.  $c_3$ 은  $o_1$ 이 존재하기 때문에 결과에서 포함되지 않지만,  $c_5$ 는 근처에  $q$ 를 지배할만한 객체가 존재하지 않으므로 리버스 스카이라인 결과에 포함된다.

### 3. 제안하는 기법

2.3절에서 소개한 거리공간에서 리버스 스카이라인 질의처리기법은 일반적인 스카이라인에 적용 가능하고 거리공간에서 처리가 가능한 효율적인 기법이다. 하지만 리버스 스카이라인 후보객체를 찾은 뒤에 최종검사를 거쳐야만 리버스 스카이라인 결과가 결정된다. 따라서 영역 안에 많은 후보객체들이 밀집하여 있다면 최종검사를 위해 많은 비용이 소모된다. 또한 연속질의처리를 수행하려면 매번 후보객체에 대한 최종검사를 수행하여야 하기 때문에 연속질의처리에 적합하지 않은 기법이다. 따라서 3장에서는 리버스 스카이라인 영역과는 별도로 객체를 검사하는 영역을 따로 설정하여 최종검사과정이 필요 없는 리버스 스카이라인 처리기법을 제안한다.

#### 3.1 스카이라인 질의처리기법

2.3절에서 소개한 기법에서 이등분선을 이용하여 리버스 스카이라인 결과에 포함되지 않는 객체를 효과적으로 필터링 했다. 하지만 이등분선으로 이루어진 리버스 스카이라인 영역 안에 있는 객체들이 리버스 스카이라인 결과라는 것을 보장하지 못했다. 때문에 후보객체로부터  $q$ 보다 가까운 위치에  $q$ 를 지배할 수 있는 객체가 존재하는지 확인하는 최종검사과정이 필요했다. 제안하는 기법은 최종검사과정을 없애기 위해 이등분선으로 이루어진 리버스 스카이라인 영역과는 별도로 객체를 검사하는 영역을 따로 설정한다. 이 영역 안에 있는 객체들을 모두 검사하여 설정된 리버스 스카이라인 영역 안에 있는 객체들은 리버스 스카이라인 결과임을 보장

하기 때문에 최종검사과정이 필요 없다. 최종검사과정이 필요하지 않기 때문에 객체의 밀집도가 높더라도 비용이 늘어나지 않으며, 리버스 스카이라인 영역에 새로운 객체가 나타나거나 사라지는 것만 모니터링 하여도 연속질의를 처리하는 것이 가능하기 때문에 기존 기법보다 효과적이다.

그림 4는 본 논문에서 제안하는 거리공간에서의 리버스 스카이라인 질의처리의 초기화 과정을 나타낸 그림이다. 먼저 그림 4(a)는 기존의 기법과 같은 방법으로  $q$ 를 지배할 수 있는 참조객체  $o_2$ 를 찾고 이등분선을 그린 그림이다. 제안하는 기법에서 수직이등분선은  $c_1, c_2$  같은 대상 객체만을 필터링 하는 것이며,  $o_1, o_3$  같은 검사해야 하는 객체는 필터링 되지 않는다. 그 다음은 그림 4(b)와 같이 다음 가까운 객체인  $o_1$ 이 검색되고 정적 속성들을 비교한 결과  $q$ 를 지배하는 참조객체라면 수직 이등분선이 그려진다. 이때 수직이등분선간에 교점이 생긴다면 교점을 중심으로 하는 객체를 검사하는 영역이 그려진다. 교점은  $q$ 를 지배할 수 있는  $o_1, o_2$  그리고 질의  $q$ 로부터 같은 거리에 위치하게 된다. 검사영역은 교점을 중심으로 하고, 교점으로부터  $q$ 까지의 거리를 반지름으로 가지는 원을 그려 검사영역으로 설정한다. 교점은 어떠한 객체가 스카이라인 영역 안에서  $q$ 로부터 가장 멀리 떨어져 있는 객체가 존재하는 상황을 가정한 것으로써, 기존 기법에서 마지막에 후보객체들을 대상으로 최종검사를 실시하는 대신 제안하는 기법에서는 처리과정에 교점을 중심으로 미리 검사를 실시하는 것이다. 즉, 리버스 스카이라인 영역 안에서  $q$ 근처에 어떤

객체가 존재한다고 하더라도 교점을 중심으로 검사하는 영역보다 큰 영역을 검사하는 객체는 존재하지 않기 때문에 교점을 중심으로 그린 영역 안에 있는 모든 객체들로부터  $q$ 를 지배하는 객체를 검색하고 리버스 스카이라인 영역이 그려진다면 그 안에 있는 결과들은 리버스 스카이라인임을 보장한다. 그럼 4(d)와 같이 흰색으로 표시된 검사영역 안의 모든 객체에 대한 검사를 마치면, 리버스 스카이라인 영역 안에 존재하는  $c_8$ 는 리버스 스카이라인 결과이다.

그림 5는 제안하는 리버스 스카이라인 초기화 알고리즘을 나타낸 의사코드이다. 입력은 그리드 색인구조[9]  $G$ 와 질의의 주체가 되는 객체  $q$ , 그리고 출력은 객체  $q$ 를 스카이라인으로 가지는 객체이다. 객체  $q$ 를 질의 테이블로부터 입력 받으면 가장 가까운 객체를 찾아서 정적속성을 비교한다. 줄 6~14는  $q$ 를 정적속성으로 지배하는 객체를 찾았을 때 이루어지는 처리과정이다. 이 중에서 줄 8~11은 수직이등분선이 교차하여 검사영역을 만들고, 객체와 셀을 필터링 하는 과정을 나타낸다. 검사영역 안의 모든 객체를 검사하고 나면 리버스 스카이라인 결과를 반환한다.

### 3.2 연속질의처리 기법

제안하는 기법은 그림 4(d)의 객체  $c_1, c_2$  같은 대상 객체들이 움직이는 환경에서 효과적으로 연속질의를 처리한다. 기존 기법은 리버스 스카이라인 영역이 존재하더라도 새로운 객체가 나타나거나 영역 안에서 객체들이 움직였을 때 결과임을 확인하기 위한 최종검사과정을 매번 수행한다. 움직인 객체가 리버스 스카이라인 영

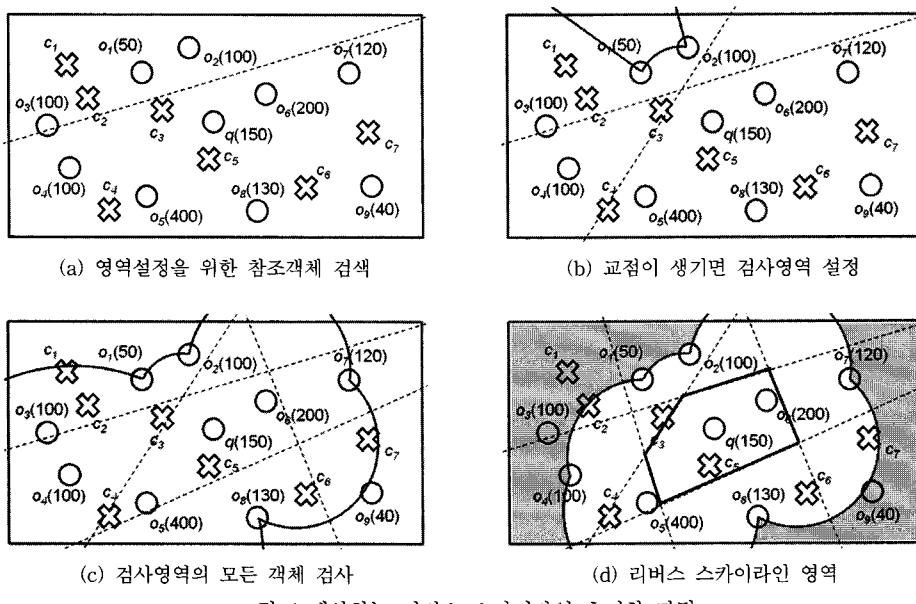


그림 4 제안하는 리버스 스카이라인 초기화 과정

---

```

ComputeReverseSkyline( $G, q$ )
input :  $G$ : 그리드 인덱스,  $q$ : 질의의 주체가 되는 객체
output : RESULT_SET : 객체  $q$ 를 스카이라인으로 가지는 객체
00 {
01   Query Table로부터 새로운  $q$ 를 입력
02    $c =$  객체  $q$ 를 포함하는 셀
03    $c$ 에 삽입
04   While ( $H$ 에 더 이상  $q$ 가 존재하지 않을 때 까지) {
05      $s = q$  와 같은 타입의 객체
06     If (  $q$  가  $s$ 에 자배당하는 경우 ) {
07        $q$  와  $s$ 의 최단거리에 대한 수직이동분선을 작성
08       If ( 두 개의 수직이동분선이 교차할 경우 ) {
09         교차점으로부터 검사영역을 작성
10          $H$ 에서 검사영역 바깥쪽에 존재하는 객체들과 셀들을 모두 제거
11       }
12     RESULT_SET에 존재하는  $q$ 와 타입이 다른 객체 중에 이동분선 바깥쪽에 존재하는 객체들을 제거
13   }
14 }
15 return RESULT_SET
16 }

```

---

그림 5 제안하는 리버스 스카이라인 초기화 알고리즘

역 안에 존재하는지 판별하는 비용이  $C_{Range}$ , 리버스 스카이라인 영역 안에 존재하는 객체들이 실제 결과인지 최종검사를 하는 비용이  $C_{Refine}$ 라면, 기존 기법을 이용하여  $n$ 번의 주기만큼 연속질의를 처리하는데 소모되는 비용은  $n * (C_{Range} + C_{Refine})$ 이다. 하지만 제안하는 기법은 초기화 단계를 거쳐 그림 4(d)와 같은 스카이라인 영역이 설정이 되면, 영역 안에 객체가 나타나는지 또는 사라지는지 모니터링 하는 것만으로 리버스 스카이라인 결과를 업데이트 하기 때문에 소모되는 비용은  $n * C_{Range}$ 이다. 제안하는 기법은 기존 기법보다  $n * C_{Refine}$ 만큼 적은 비용으로 리버스 스카이라인 연속질의처리를 수행한다.

그림 6은 연속질의처리를 위한 공간색인구조를 보여주는 그림이다. 질의는  $\langle qid, coordinates, reference object, result set \rangle$ 의 정보를 유지한다.  $coordinates$ 는 질의의 좌표이며,  $reference object$ 는 질의의 리버스 스카이라인 영역을 만드는데 참조한 객체들의 리스트이다.  $result set$ 에는 질의의 리버스 스카이라인 결과가 저장되어 있다. 객체는  $\langle oid, coordinates, type, values \rangle$ 의 정보를 유지한다.  $values$ 는 객체가 가지는 속성값들이다.  $type$ 은 질의와 같은 타입의 객체인지 다른 타입의 객체인지를 나타낸다. 각 셀은 존재하는 객체의 리스트와 셀에 영향을 주는 질의의 리스트를 유지하여 리버스 스카이라인 연속질의를 처리한다.

그림 7은 제안하는 리버스 스카이라인 연속질의처리 알고리즘을 나타낸 의사코드이다. 입력은 그리드 색인구조[9]  $G$ 와 이동객체  $o$ 이다. 객체  $o$ 가 과거위치에서 현재위

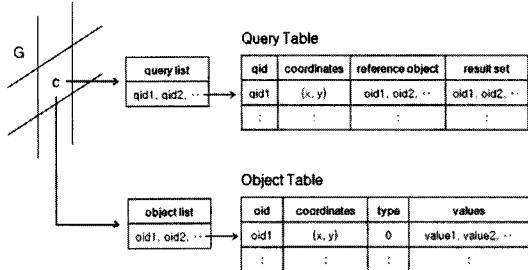


그림 6 연속질의처리를 위한 공간색인구조

---

```

ComputeContinuousReverseSkyline( $G, o$ )
input :  $G$ : 그리드 인덱스,  $o$ : 이동 객체
00 {
01    $o$ 가 이전 위치로부터 현재위치로 이동
02    $q/list =$  이전위치와 현재위치를 포함하는 셀에 등록된 질의리스트
03   While ( $q/list$ 에 질의가 존재하지 않을 때 까지) {
04      $q = q/list$ 에 포함된 질의
05      $s1 =$  이전 위치에서  $q$ 의 RESULT_SET에 포함되는지 상태유무
06      $s2 =$  현재 위치에서  $q$ 의 RESULT_SET에 포함되는지 상태유무
07     If (  $s1$  과  $s2$ 의 상태가 다르다면 ) {
08       If (  $s1$  이  $q$ 의 리버스 스카이라인 영역 안에 존재하면 ) {
09          $q$ 의 RESULT_SET에  $o$ 를 추가
10       } else if (  $s1$  이  $q$ 의 리버스 스카이라인 영역 안에 존재하지 않으면 )
11          $q$ 의 RESULT_SET에서  $o$ 를 제거
12     }
13   }
14 }

```

---

그림 7 제안하는 리버스 스카이라인 연속질의처리 알고리즘

치로 이동을 하면,  $o$ 의 과거위치와 현재위치에 해당하는 셀로부터 질의의 리스트를 불러온다. 그 다음 질의들의 결과 셋으로부터  $o$ 가 과거에 각 질의의 결과에 포함되어 있었는지, 포함되지 않았았는지 상태를 저장한다. 그리고 현재위치로 질의들의 리버스 스카이라인 영역에 포함되는지, 포함되지 않는지 상태를 계산하여 저장한다. 리버스 스카이라인 영역은 저장된 참조객체 리스트를 이용하여 계산한다. 이전상태와 현재상태를 비교하여 차이가 존재하면 해당 질의의 결과를 업데이트한다. 이와 같은 방법으로 객체가 이동하는 환경에서 효과적으로 연속질의처리를 수행한다.

#### 4. 성능평가

##### 4.1 실험환경

본 논문에서 제안하는 기법의 성능평가는 Pentium 3.0GHz 프로세서와 1GB의 메인 메모리를 가진 Windows XP 운영체계를 사용하는 데스크탑에서 구현하였다. 기본 테스트 환경으로 이동객체 거리공간의 좌표 범위는  $10,000 \times 10,000$ 로 가정했고, 객체의 개수는 10,000개 그리고 셀의 개수는  $20 \times 20$ 개로 했다. 질의는 각각 1,000개씩 수행하였으며, 질의와 객체의 위치 속성과 정적 속성은 모두 랜덤 함수를 이용했다.

리버스 스카이라인 질의처리 기법의 성능을 평가하기 위해서 2.3절에서 소개한 수직이동분선만을 이용하는 기존기법[8]과 제안하는 기법으로 질의가 처리되는 동안 셀에 접근하는 횟수를 비교 측정했다. 성능평가는 질의의 속성값, 셀의 개수를 변화시켜가며 어떻게 셀 접근횟수가 변화하는지 비교 분석하는 방법으로 실시됐다. 그리고 기존 기법과 제안하는 기법을 이용하여 연속질의가 처리되는 동안 셀에 접근하는 횟수를 측정하여 비교 분석했다.

##### 4.2 성능평가 결과

그림 8은 다른 환경이 동일한 상태에서 질의의 속성값을 1000에서 10000까지 변화시켜가며 리버스 스카이라인 질의처리에 따른 셀의 접근 횟수를 나타낸 그래프이다. 질의의 속성값을 변화시켜가며 성능을 평가해본

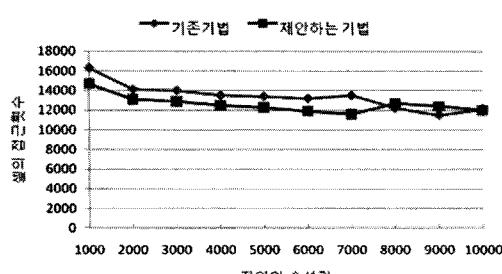


그림 8 질의의 속성값 변화에 따른 성능 비교

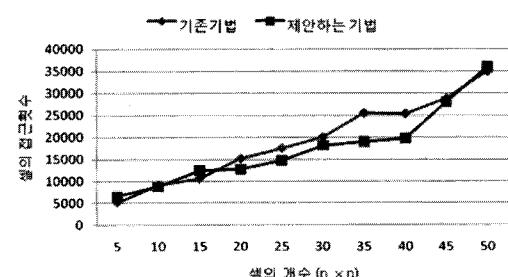


그림 9 셀의 개수 변화에 따른 성능 비교

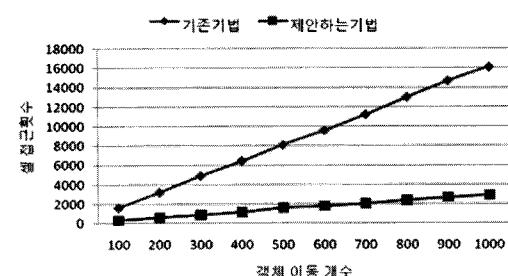


그림 10 이동하는 객체의 개수 변화에 따른 성능 비교

결과 속성값이 작을 때는 제안하는 기법이 기존 기법보다 우수한 성능을 보였다. 속성값이 작으면 리버스 스카이라인 결과가 많기 때문에 최종검사를 수행하는 기존 기법을 처리하는 비용이 더 크다. 하지만 질의의 속성값이 커서 리버스 스카이라인 결과가 적은 경우에는 최종 검사의 수행이 적어서 셀의 접근횟수가 비슷하거나 제안하는 기법의 셀 접근횟수가 많았다.

그림 9는 다른 환경이 동일한 상태에서 셀의 개수를  $5 \times 5$ 에서  $50 \times 50$ 까지 변화시켜가며 리버스 스카이라인 질의처리에 따른 셀의 접근 횟수를 나타낸 그래프이다. 셀의 개수를 변화시켜가며 성능을 평가해본 결과 제안하는 기법이 기존기법보다 우수한 성능을 보였다. 셀의 개수에 따른 변화는 셀의 개수가 많으면 하나의 셀에 들어있는 객체의 수가 적으므로 보다 많은 셀에 접근하여 리버스 스카이라인 질의를 처리한다.

이 실험에서는 리버스 스카이라인 연속질의처리 성능을 평가했다. 그림 10은 이동하는 객체의 수를 변화시켜가며 연속질의가 처리되는 동안 셀의 접근 횟수를 나타낸 그래프이다. 기존 기법은 매번 최종검사를 하기 때문에 셀 접근횟수가 매우 많다. 하지만 제안하는 기법은 최종검사과정이 없기 때문에 적은 비용으로 연속질의처리를 수행한다. 따라서 제안하는 기법을 사용하면 그림 8과 그림 9에서 나타나는 것처럼 초기화 과정에서는 일부 더 많은 비용이 드는 경우가 생길 수 있다. 그러나 연속질의처리를 하는 과정에서 비용이 적게 들기 때문에 연속 질의처리에는 제안한 기법이 기존 기법보다 효과적이다.

## 5. 결 론

본 논문에서는 일반적인 스카이라인에 적용 가능하고 거리공간에서 처리가 가능한 리버스 스카이라인 처리기법을 제안했다. 제안하는 기법은 스카이라인에서 객체의 지배관계와 이동분선의 성질을 이용하여 거리공간에서 효율적으로 리버스 스카이라인 질의를 처리한다. 또한 제안하는 기법을 이용하여 연속적인 질의를 처리하기 위해, 최종검사 단계에서 발생하는 영역질의를 줄일 수 있는 리버스 스카이라인 질의처리기법을 제안한다. 제안하는 기법의 우수성을 보이기 위해서 제안하는 기법과 기존에 연구된 리버스 스카이라인 질의처리기법에 대한 성능평가를 수행하고 비교 분석했다. 그 결과, 기존의 기법보다 제안하는 기법이 연속질의를 처리하는데 우수한 성능을 보이는 것을 확인했다. 향후 연구로는 모든 객체가 움직이는 환경을 고려하여 리버스 스카이라인 연속질의 처리기법을 연구할 것이다.

## 참 고 문 헌

- [1] S. Borzsonyi, D. Kossmann, and K. Stocker, "The Skyline Operator," *Proc. of the IEEE International Conference on Data Engineering*, pp.421-430, 2001.
- [2] C. Li, B. B. Ooi, A. K. H. Tung, and S. Wang, "DADA: a Data Cube for Dominant Relationship Analysis," *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp.659-670, 2006.
- [3] Z. Li, Y. Park, "Efficient Processing using Static Validity Circle for Continuous Skyline Queries," *Journal of KIISE : Databases*, vol.33, no.6, pp.631-643, 2006. (in Korean)
- [4] E. Dellis, and B. Seeger, "Efficient Computation of Reverse Skyline Queries," *Proc. of the International Conference on Very Large Data Bases*, pp.291-302, 2007.
- [5] A. Han, J. Kim, D. Ahn, Y. Park, "Efficient Reverse Skyline Query Processing of Companies Perspective," *Proc. of the KIISE Fall Conference*, vol.35, no.2(C), pp.49-54, 2008. (in Korean)
- [6] X. Lian, and L. Chen, "Monochromatic and Bichromatic Reverse Skyline Search over Uncertain Databases," *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp.213-226, 2008.
- [7] L. Zhu, C. Li, and H. Chen, "Efficient Computation of Reverse Skyline on Data Stream," *Proc. of the International Joint Conference on Computational Sciences and Optimization*, vol.1, pp.735-739, 2009.
- [8] J. Lim, Y. Park, D. Seo, J. Lee, S. Jang, and J. Yoo, "Reverse Skyline Query Processing in Metric Spaces," *Journal of KIISE : Computing Practices and Letters*, vol.16, no.7, pp.809-813, 2010. (in Korean)
- [9] M. F. Mokbel, X. Xiong, and W. G. Aref, "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases," *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp.623-634, 2004.

### Korean)

- [9] M. F. Mokbel, X. Xiong, and W. G. Aref, "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases," *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp.623-634, 2004.

### 임 종 태

2009년 2월 충북대학교 정보통신공학과 공학사. 2009년 3월~현재 충북대학교 정보통신공학과 석사과정. 관심분야는 데이터베이스 시스템, 무선 센서 네트워크, 위치기반 서비스



### 박 용 훈

2005년 2월 호원대학교 정보통신공학과 공학사. 2007년 2월 충북대학교 정보통신공학과 공학석사. 2009년 2월~현재 충북대학교 정보통신공학과 박사과정. 관심분야는 데이터베이스 시스템, 정보검색, 시공간 데이터베이스, 파일시스템, 위치기반서비스



### 서 동 민

2002년 2월 충북대학교 정보통신공학과 공학사. 2004년 2월 충북대학교 정보통신공학과 공학석사. 2008년 2월 충북대학교 정보통신공학과 공학박사. 2008년 2월~2010년 2월 한국과학기술원 정보전자연구소 연구원. 2010년 2월~현재 한국과학기술정보연구원 정보기술연구실 연구원. 관심분야는 이동객체 데이터베이스 시스템, XML 데이터베이스 시스템, 무선 센서 네트워크, 시멘틱 웹 서비스(RDF, OWL 등)



### 유 재 수

1989년 2월 전북대학교 컴퓨터공학과 공학사, 1991년 2월 한국과학기술원 전산학과 공학석사, 1995년 2월 한국과학기술원 전산학과 공학박사, 1995년 3월~1996년 8월 목포대학교 전산통계학과 전임강사, 1996년 8월~현재 충북대학교 전기전자컴퓨터공학부 및 컴퓨터정보통신연구소 교수. 관심분야는 데이터베이스시스템, 정보검색, 센서네트워크 및 RFID, 멀티미디어 데이터베이스, 분산객체컴퓨팅

