거리공간에서의 리버스 스카이라인 질의 처리

(Reverse Skyline Query Processing in Metric Spaces)

임 종 태 † 박 용 훈 †
(Jongtae Lim) (Yonghun Park)
서 동 민 †† 이 진 주 †
(Dongmin Seo) (Jinju Lee)
장 수 민 † 유 재 수 †††
(Soomin Jang) (Jaesoo Yoo)

요 약 최근의 기업중심의 서비스를 위해서 리버스 스카이라인 질의 처리가 연구되었다. 하지만 지금까지의 리버스 스카이라인에 대한 연구는 모두 다이나믹 스카이라인을 기반으로 한 리버스 다이나믹 스카이라인이고, 위치 기반 서비스를 위한 거리공간에서의 리버스 스카이라인 질의 처리 기법은 전무하다. 따라서 본 논문에서는 일반적인 스카이라인에 적용 가능하고 거리공간을 고려한 리버스 스카이라인 처리 기법을 제안한다. 제안하는 기법은 기존의 공간 색인을 활용하여 거리공간에서 리버스 스카이라인을 처리하며, 객체의 단색적인 환경과 양색적인 환경을 모두 고려한다. 제안하는 기법의 우수성을 보이기 위해 제안하는 기법과 기본적인 리버스 스카이라인 질의 처리 기법과의 성능 평가를 수행하고 그 결과를 비교 분석했다. 그 결과 기존의 기법보다 약 5000배 우수한 성능을 보였다.

키워드 : 리버스 스카이라인, 거리공간, 위치 기반 서비스

Abstract Many studies on reverse skyline query processing have been done for company oriented services. The existing methods about reverse skyline are reverse skyline based on dynamic skyline. There is no reverse skyline query processing algorithm based on metric spaces for location-based services. In this paper we propose a reverse skyline query processing scheme that applies for a general skyline and considers distance spaces. The proposed method processes reverse skyline queries in the metric spaces using the existing spatial indexing scheme and considers both Monochromatic and Bichromatic environments. In order to show the superiority of the proposed scheme, we compare it with the basic skyline query processing scheme through performance evaluation. As a result, the proposed method excellent performance was about 5000 times more than conventional method.

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

1. 서 론

최근 위치 인식 기술(Location Aware Technology: LAT) 및 무선 인터넷 기술의 발전과 더불어 휴대폰, PDA, 노트북 같은 모바일 기기의 보급이 확산되면서 고객의 위치를 확인하고 주변정보를 이용하여 고객에게 유용한 정보를 제공하는 위치 기반 서비스(Location-Based Service: LBS)가 다양하게 연구되어 왔다. 위치 기반 서비스에서 이동 객체(Moving Object)의 질의를 처리하는 대표적인 연구로는 포인트 질의, 영역 질의, kNN(k-Nearest Neighbor) 그리고 Top-k 등이 있다.

이런 이동 객체 관련 질의 처리는 점차 정보에 대한 고객들의 요구가 다양해지면서 한 가지 속성만을 고려하는 방법으로는 고객들의 요구를 충족하기가 힘들어지게 되었다. 그래서 이를 해결하기 위한 다중 속성을 동시에 고려할 수 있는 스카이라인 질의 처리(Skyline Query)가 연구되었다[1-6]. 스카이라인은 질의의 요구에 적합한 객체를 검색하는 질의 처리 기법으로 전체 객체 집합으로부터 다른 객체들에 의해 지배당하지 않는 질의가 선호할만한 객체들로 이루어진 부분집합을 검색한다. 이러한 스카이라인이 연구됨에 따라서 고객은 거리가 가깝고, 가격이 저렴한 음식점 같은 정보를 검색하여 손쉽게 유용한 정보를 얻을 수 있게 되었다.

최근에는 기업 입장에서 서비스를 제공하는 방법에 대해서도 많은 연구를 하고 있다. 기존에는 '어떤 정보

· 본 연구는 2009년 교육과학기술부(지역거점연구단육성사업/충북BIT연구중심대학육성사업단)와 한국산업기술재단의 지역혁신인력양성사업으로터 지원받아 수행된 연구 결과임
· 이 논문은 제36회 추계학술발표회에서 '거리공간에서의 리버스 스카이라인 질의 처리'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 충북대학교 정보통신공학과
jtlim@chungbuk.ac.kr
yhpark1119@chungbuk.ac.kr
jinjulee83@chungbuk.ac.kr
jsm@chungbuk.ac.kr
†† 정 회 원 : 한국과학기술정보연구원 정보기술연구실
dmseo@kisti.re.kr
††† 종신회원 : 충북대학교 정보통신공학과 교수
yjs@chungbuk.ac.kr
(Corresponding author임)

논문접수 : 2009년 12월 28일
심사완료 : 2010년 3월 29일

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 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제7호(2010.7)

를 제공할 것인가?'와 같이 정보의 유형만을 고려한 서비스였다면, 최근에는 어떤 정보를 어떤 대상에게 어떤 방법으로 효과적으로 전달할 것인가?'와 같이 대상이나 방법도 중요하게 고려되는 서비스가 요구되고 있다. 스카이라인은 고객이 주체가 되어 정보를 검색하고 필요한 정보를 얻는 기존의 서비스에는 유용한 질의 처리 기법이지만 기업 중심의 서비스를 처리할 때는 다소 적합하지 못한 기법이다. 그래서 연구된 질의 처리 기법이 리버스 스카이라인 질의 처리(Reverse Skyline Query)이다[7-10].

리버스 스카이라인은 질의의 주체가 되는 객체를 스카이라인으로 여기는 객체를 검색하는 질의 처리 기법이다. 리버스 스카이라인이 연구됨에 따라서 기업 입장에서 '우리 기업을 가격이 저렴하고, 거리를 가깝게 여기는 고객'들을 검색할 수 있게 되었고, 이런 정보들을 활용하여 과거보다 유용한 서비스를 제공할 수 있게 되었다. 지금까지 리버스 스카이라인에 대한 연구는 모두 다이나믹 스카이라인(Dynamic Skyline)을 기반으로 한 리버스 다이나믹 스카이라인 질의 처리(Reverse Dynamic Skyline Query)에 대한 연구였다. 하지만 리버스 스카이라인을 이용한 다양한 서비스를 제공하기 위해서는 지금까지 연구된 리버스 다이나믹 스카이라인만으로는 한계가 있으며, 거리공간(Metric Space)에서의 리버스 스카이라인 처리 기법 또한 전무하다.

본 논문에서는 거리공간에서 일반적인 리버스 스카이라인을 처리하기 위한 기법을 제안한다. 제안하는 기법은 기존의 공간색인을 활용하여 거리공간에서 리버스 스카이라인을 처리하였으며, 객체의 단색적인(Monochromatic) 환경과 양색적인(Bichromatic) 환경을 모두 고려하였다. 그리고 제안하는 기법과 기본적인 리버스 스카이라인 처리 기법과의 성능평가를 수행하고 그 결과를 비교 분석했다. 성능평가를 통해 제안하는 기법이 기존의 GRSL 기법[10]에 비해 셀 접근 횟수 측면에서 최고 약 5,000배 우수한 성능이 나오는 것을 확인했다.

본 논문의 구성은 아래와 같다. 2장에서는 관련연구를 기술한다. 3장에서는 제안하는 질의 처리 기법을 기술하고 4장에서는 성능 평가를 통해 그 우수성을 보여준다. 5장에서 결론과 향후 연구를 기술하고 본 논문의 기술을 마친다.

2. 관련연구

2.1 리버스 스카이라인(Reverse Skyline)

리버스 스카이라인이란 질의의 주체가 되는 객체를 스카이라인으로 여기는 객체를 검색하는 질의 처리 기법이다. 예를 들면 상점의 입장에서 주변에 위치한 고객들 중에 본 상점을 가격이 저렴하고, 거리가 가깝다고 여기는 고객을 대상으로 광고를 하려고 한다면 리버스 스카이라인이 필요하다.

리버스 스카이라인의 가장 기본적인 처리 기법인 GRSL (Greedy Reverse Skyline)[7]은 모든 객체에 대해서 스카이라인을 구하고 질의의 주체가 되는 객체가 객체들의 스카이라인 결과에 포함되어 있는지 확인하는 기법이다. 하지만 이 기법은 모든 객체에 대해서 스카이라인을 수행하여야 하기 때문에 처리 시간과 비용이 많이 든다.

리버스 스카이라인을 효과적으로 처리하기 위해 많은 기법이 제안되었다. 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 소개되었으며[7], RSSA에서 문제가 되었던 DSL의 선 저장과 영역질의를 사용하지 않는 ERSL (Efficient Reverse Skyline)이 제안됐다[8]. 또한 일정한 범위를 가진 확실하지 않는 데이터에 대한 리버스 스카이라인 처리 기법도 제안되었으며[9], 슬라이딩 윈도우 기법을 이용하여 리버스 스카이라인의 스트림 데이터를 처리하는 기법도 제안됐다[10].

3. 거리공간에서의 리버스 스카이라인 질의 처리

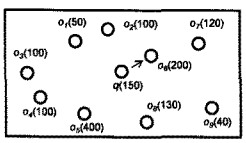
논리적인 공간에서의 다이나믹 리버스 스카이라인 질의 처리만으로는 다양한 위치 기반 서비스를 제공하기 어렵기 때문에 거리공간에서의 리버스 스카이라인이 필요하다. 제안하는 기법은 기존의 공간색인 기법을 활용하여 거리공간에서 리버스 스카이라인을 처리했으며, 객체의 단색적인 환경과 양색적인 환경을 모두 고려하였다.

3.1 단색적인 환경에서의 리버스 스카이라인

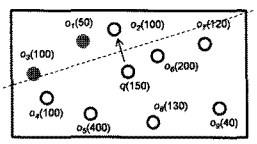
단색적인(Monochromatic) 환경이란 질의의 주체가 되는 객체와 질의의 대상이 되는 객체들이 동일한 환경이다. 모든 객체들은 위치 속성과 정적 속성을 가진다. 예를 들면 서바이벌 게임 중에 질의가 되는 주체는 나 자신이 되고, 대상이 되는 객체들은 다른 서바이벌 게임 참가자이다. 모든 객체는 총알 개수와 위치라는 속성을 가지고, 총알 개수가 적고 거리가 가까운 다른 참가자를 공격하는 것이 유리하다. 이때 현재 나를 공격할 만한 사람을 알고 싶다고 한다면 다른 참가자들이 총알 개수와 거리로 스카이라인을 처리한 결과 나를 스카이라인으로 가지는지 확인해야 한다. 이런 질의를 처리하기 위한 방법으로 단색적인 환경에서의 리버스 스카이라인이 필요하다.

그림 1은 단색적인 환경에서의 리버스 스카이라인 질의 처리 과정을 나타낸 그림이다. 먼저 그림 1(a)는 위

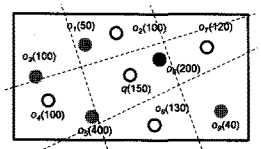
치 속성에 의해 객체들이 거리공간에 위치한 그림으로 괄호 안에 있는 수치는 객체의 정적 속성이다. 그림과 같이 위치하고 나면 질의의 주체가 되는 객체 q는 가까운 객체들로부터 거리가 아닌 속성으로 q를 지배하는 객체를 검색한다. 그림 1(a)와 같이 최초로 o_6을 선택하고 정적 속성들을 비교한 결과 q가 지배를 당하지 않으면 o_6은 다른 객체들이 q를 스카이라인으로 여기는데 영향을 주지 못하는 객체이므로 고려대상에서 제외되며 o_6은 거리상으로 가까운 q를 스카이라인으로 여기기 때문에 리버스 스카이라인 결과에 포함된다. 그림 1(b)와 같이 다음 가까운 객체인 o_2와 비교한다. 정적 속성들을 비교한 결과 q를 지배하는 객체라면 두 객체의 최단거리를 수직이등분 하는 선을 긋는다. q를 기준으로 이등분선 바깥쪽에 위치하는 객체 o_1, o_3의 스카이라인에서 q는 거리, 정적 속성 모두 o_2의 지배를 받기 때문에 o_1, o_3은 리버스 스카이라인의 결과에 포함될 수 없다. 이런 방법으로 가까운 객체부터 차례로 검색해 나가면 그림 1(c)와 같이 영역을 설정할 수 있다. 영역이 설정되면 q를 지배하는 객체들을 제외한 나머지 객체들은 결과에 포함되는지 아닌지 결정된다. 마지막으로 그림 1(d)와 같이 q를 지배하는 객체들의 주변에 q를 지배하는 다른 객체가 있는지 확인하는 검사를 마치면 q에 대한 리버스 스카이라인의 처리가 모두 끝나게 된다. o_2는 o_1, o_4는 o_3, o_7은 o_9가 있어서 결과에서 포함되지 않지만, o_8의 경우 근처에 q를 지배할만한 객체가 존재하지 않으므로 결과에 포함된다. 최종검사를 마치면 o_6, o_8이 q에 대한 리버스 스카이라인의 최종결과이다. 그림 2는 단색적인 환경에서의 리버스 스카이라인 처리 알고리즘을 나타낸 의사코드(Pseudo code)이다. 입력은 질의의 주체가 되는 객체 q, 그리고 출력은 객체 q를 스카이라인으로 가지는 객체이다.



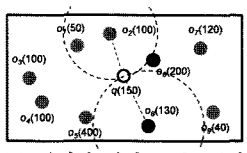
(a) 가까운 객체부터 지배 가능한 객체를 검색



(b) 지배하는 객체를 만나면 이등분선을 그림



(c) 영역이 설정될 때까지 검색



(d) 선택된 객체를 최종검사

그림 1 단색적인 환경에서의 리버스 스카이라인 처리 과정

알고리즘 1. 단색적인 환경에서의 리버스 스카이라인

```
입력 : 질의의 주체가 되는 객체 q
출력 : 객체 q를 스카이라인으로 가지는 객체
01: 객체 q를 포함하는 셀을 queue에 push
02: while (queue에 셀이 존재하지 않을 때 까지)
03:   queue에서 질의로부터 가장 가까운 셀을 pop
04:   if (pop된 셀의 레벨이 n이면)
05:     레벨이 n+1인 셀들을 queue에 push
06:   end if
07:   while (pop된 셀 안의 객체가 모두 비교될 때까지)
08:     pop된 셀 안의 객체 중 가까운 객체부터 정적 속성 비교
09:     if (객체가 q와 정적 속성이 같거나 q를 지배하는 객체라면)
10:       객체와 q의 최단거리를 이등분하는 선을 그림
11:       q를 기준으로 선 바깥쪽에 존재하는 셀들을 queue에서 걸러냄
12:       q를 기준으로 선 바깥쪽에 존재하는 객체들을 걸러냄
13:       객체를 리버스 스카이라인 결과의 후보로 저장
14:     else if (객체가 q에게 정적 속성이 지배당하는 객체라면)
15:       객체를 리버스 스카이라인의 결과로 저장
16:     end if
17:   end while
18: end while
19: 리버스 스카이라인 결과의 후보로 저장된 객체들을 불러옴
20: if (객체와 q의 거리보다 가까운 위치에 q와 정적 속성이 같거나
21:   q를 지배하는 객체가 존재하지 않으면)
22:   객체를 리버스 스카이라인의 결과로 저장
23: end if
24: 걸러져야 하는데 걸러지지 않은 객체가 존재하는지 최종검사
25: return 리버스 스카이라인의 결과에 저장된 객체들
```

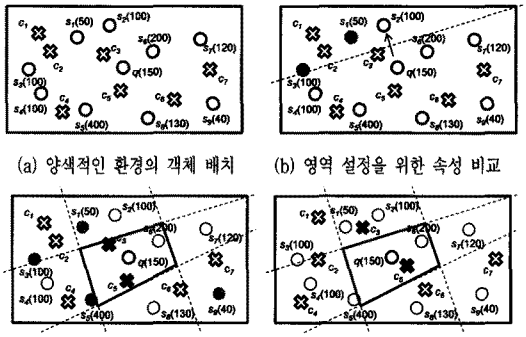
그림 2 단색적인 환경에서의 리버스 스카이라인 처리 알고리즘

3.2 양색적인 환경에서의 리버스 스카이라인

양색적인(Bichromatic) 환경이란 질의의 주체가 되는 객체와 질의의 대상이 되는 객체들이 서로 다른 환경이다. 질의의 주체가 되는 객체들과 같은 성질을 가진 객체들은 위치 속성과 정적 속성을 가지며, 질의의 대상이 되는 객체들은 위치 속성을 가진다. 예를 들면 질의의 주체가 되는 객체는 상점, 질의의 대상이 되는 객체는 고객이다. 상점은 가격과 위치라는 속성을 가지고, 고객은 위치 속성을 가진다. 상점 입장에서 본 상점을 가격이 저렴하고 거리를 가깝게 여기는 고객들을 대상으로 광고를 하여 광고의 효율을 높이고자 한다. 이러한 질의를 처리하기 위한 방법으로 양색적인 환경에서의 리버스 스카이라인이 필요하다.

그림 3은 양색적인 환경에서의 리버스 스카이라인 질의 처리 과정을 나타낸 그림이다. 먼저 그림 3(a)는 위치 속성에 의해 두 가지의 서로 다른 객체들이 거리공간에 위치한 그림이다. 질의의 주체가 되는 객체 q는 그림 3(b)와 같이 자신과 같은 속성을 가진 s_1, s_2, s_3 같은 객체들을 대상으로 자신을 지배하는 객체를 검색

한다. 단색적인 환경에서의 리버스 스카이라인과 동일한 기법으로 그림 3(c)와 같이 영역을 설정하고 나면 영역 안에 있는 대상 객체인 c_3, c_5가 객체 q의 리버스 스카이라인 결과가 된다. 그림 3(d)는 대상 객체들이 움직이는 환경을 보여주는 그림이다. 영역이 설정되고 나면 영역 안에 객체들이 들어오는지 나가는지 관찰함으로써 대상 객체들이 움직이는 환경에서도 적은 업데이트 비용으로 리버스 스카이라인 질의를 처리한다. 그림 4는 양색적인 환경에서의 리버스 스카이라인 처리 알고리즘을 나타낸 의사코드이다. 입력과 출력은 단색적인 환경에서의 리버스 스카이라인 처리 알고리즘과 유사하다.



(a) 양색적인 환경의 객체 배치 (b) 영역 설정을 위한 속성 비교

(c) 영역이 설정될 때까지 검색 (d) 영역관찰로 이동 객체 검사

그림 3 양색적인 환경에서의 리버스 스카이라인 처리 과정

알고리즘 2. 양색적인 환경에서의 리버스 스카이라인

```
입력 : 질의의 주체가 되는 객체 q
출력 : 객체 q를 스카이라인으로 가지는 객체
01: 객체 q를 포함하는 셀을 queue에 push
02: while (queue에 셀이 존재하지 않을 때 까지)
03:   queue에서 질의로부터 가장 가까운 셀을 pop
04:   if (pop된 셀의 레벨이 n이면)
05:     레벨이 n+1인 셀들을 queue에 push
06:   end if
07:   while (pop된 셀 안의 객체가 모두 비교될 때까지)
08:     pop된 셀 안의 객체 중 가까운 객체부터 정적 속성 비교
09:     if (pop된 셀 안의 객체 중 q와 타입이 같으면)
10:       if (객체가 q와 정적 속성이 같거나 q를 지배하는 객체라면)
11:         객체와 q의 최단거리를 이등분하는 선을 그림
12:         q를 기준으로 선 바깥쪽에 존재하는 셀들을 queue에서 걸러냄
13:         q를 기준으로 선 바깥쪽에 존재하는 객체들을 걸러냄
14:       end if
15:     end if
16:   end while
17: end while
18: 걸러져야 하는데 걸러지지 않은 객체가 존재하는지 최종검사
19: q와 타입이 다른 객체들 중에 걸러지지 않은 객체들을 결과로 저장
20: return 리버스 스카이라인의 결과에 저장된 객체들
```

그림 4 양색적인 환경에서의 리버스 스카이라인 처리 알고리즘

3장에서는 제안하는 기법을 이해하기 쉽도록 위치 속성과 정적 속성 하나만을 가진 환경으로 예를 들었지만 본 논문에서 제안하는 기법은 보다 많은 속성들이 존재해도 적용 가능하다.

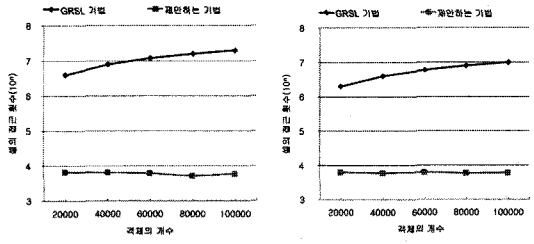
4. 성능평가

4.1 실험환경

본 논문에서 제안하는 기법의 성능평가는 Pentium 3.0GHz 프로세서와 1GB의 메인 메모리를 가진 Windows XP 운영체제를 사용하는 데스크탑에서 구현하였다. 기본 테스트 환경으로 이동객체 거리공간의 좌표 범위는 15,000×15,000로 가정했고, 객체의 개수는 100,000개 그리고 셀의 개수는 20×20개로 했다. 질의는 각각 2,000개씩 수행하였으며, 질의와 객체의 위치 속성과 정적 속성은 모두 랜덤 함수를 이용했다. 리버스 스카이라인의 성능을 평가하기 위해서 질의가 처리되는 동안 셀에 접근하는 횟수를 측정했다.

4.2 객체 개수 변화에 따른 성능 평가

이 실험에서는 다른 환경이 동일한 상태에서 객체의 개수를 20,000개부터 100,000개까지 변화시켜가며 성능을 평가했다. 그림 5는 단색적인 환경과 양색적인 환경에 따른 성능을 나타낸다. 객체의 개수를 변화시켜가며 성능을 평가해본 결과 제안하는 기법이 GRSL 기법보다 최고 약 3500배 우수한 성능을 보였다. 객체의 변화에 따른 변화는 객체가 적을수록 셀 접근 횟수가 많은 경향을 보인다. 객체의 개수가 적으면 각 셀 안에 존재하는 객체의 개수가 감소함으로써 보다 많은 셀을 뒤져야 리버스 스카이라인 처리가 가능하기 때문이다.

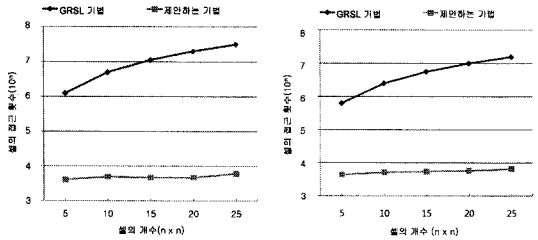


(a) 단색적인 환경 (b) 양색적인 환경

그림 5 객체 개수 변화에 따른 성능 비교

4.3 셀의 개수 변화에 따른 성능 평가

이 실험에서는 다른 환경이 동일한 상태에서 셀의 개수를 5×5부터 25×25까지 변화시켜가며 성능을 평가했다. 그림 6은 단색적인 환경과 양색적인 환경에서 성능을 나타낸다. 성능평가 결과 제안하는 기법이 GRSL 기법보다 최고 약 5000배 우수한 성능을 보였다. 제안하는 기법은 셀의 개수가 증가하면 셀의 접근 횟수가 늘어나



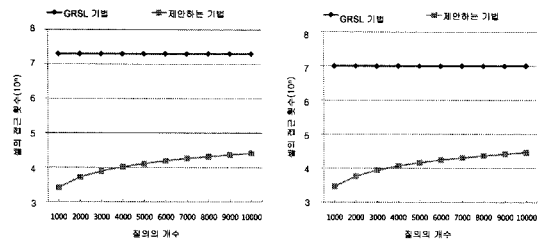
(a) 단색적인 환경 (b) 양색적인 환경

그림 6 셀의 개수 변화에 따른 성능 비교

는 경향을 보인다. 셀의 개수가 늘어나면 각 셀에 들어가는 객체의 개수가 감소함으로써 더 많은 셀에 접근해야 리버스 스카이라인을 처리할 수 있기 때문이다.

4.4 질의 개수 변화에 따른 성능 평가

이 실험에서는 다른 환경이 동일한 상태에서 질의의 개수를 1,000개부터 10,000개까지 변화시켜가며 성능을 평가했다. 그림 7은 단색적인 환경과 양색적인 환경에 따른 성능을 나타낸다. GRSL 기법은 질의를 수행하기 위해서 모든 객체의 스카이라인을 구하기 때문에 질의가 한꺼번에 여러 개가 수행이 되어도 구축된 스카이라인으로 처리가 가능하여 셀 접근 횟수가 늘어나지 않지만 큰 비용이 소모될 뿐만 아니라 이동객체 환경에서는 적합하지 않다. 제안하는 기법은 각각의 질의에 대해 위치 속성이 변화하여 새롭게 질의를 처리하여야 하기 때문에, 셀 접근 횟수가 조금씩 증가한다. 하지만 이 기법은 적은 비용을 소모할 뿐만 아니라 이동객체 환경에 적합한 기법이기 때문에 우수한 리버스 스카이라인 처리 기법이다.



(a) 단색적인 환경 (b) 양색적인 환경

그림 7 질의 개수 변화에 따른 성능 비교

5. 결 론

본 논문에서는 거리공간에서 공간색인을 활용한 효율적인 리버스 스카이라인 질의 처리 기법을 최초로 제안했다. 제안하는 기법은 가까운 대상 객체부터 정적 속성을 비교하여 질의의 주체가 되는 객체를 지배하는 객체를 검색해 영역을 만들고 추가적인 연산을 통해 질의의 주체가 되는 객체가 지배당하는 조건을 찾아 리버스 스카이라인을 처리하는 방법이며, 객체의 단색적인 환경과 양색적인 환경을 모두 고려했다. 객체의 개수, 셀의 개수, 질의의 개수를 변화시켜가며 제안하는 기법과 GRSL 기법과의 성능평가를 수행한 결과 제안하는 기법이 최고 약 5000배 우수한 성능을 보였다. 추후 연구는 객체가 움직이는 환경을 고려하여 거리공간에서 리버스 스카이라인의 연속질의 처리 기법을 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] J. Kim, Y. Park, "A Progressive Skyline Region Decision Method," *Journal of KIISE : Databases*, vol.34, no.1, pp.70-83, 2007. (in Korean)

[4] 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)

[5] D. Fuhry, R.Jin, and D.Zhang, "Efficient Skyline Computation in Metric Space," *Proc. of the International Conference on Extending Database Technology : Advances in Database Technology*, pp.1042-1051, 2009.

[6] L. Chen, X. Lian, "Efficient Processing of Metric Skyline Queries," *IEEE Transactions On Knowledge and Data Engineering*, vol.21, no.3, pp.351-365, 2009.

[7] 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.

[8] 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)

[9] 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.

[10] 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.