

고정 그리드 인덱싱에서 VP 필터링을 이용한 범위 질의 처리

전세길, 나연목
단국대학교 전자.컴퓨터공학과
sgjeon@dankook.ac.kr, ymna@dku.edu

Range Query Processing using VP Filtering in Fixed Grid Index Structures

Segil Jeon, Yunmook Nah
Dept. of Electronics & Computer Engineering, Dankook University

요 약

최근 들어 이동 통신 서비스에서 이동하는 고객의 위치 정보와 관련된 서비스가 중요한 서비스로 부각되고 있다. 이동객체의 경우 갱신 연산이 많고, 부하가 특정 지역에 집중되는 특징이 있다. 이러한 위치 기반 서비스에서 범위질의는 중요한 질의 중 하나이다. 범위 질의는 윈도우나 원 형태로 수행되는데 윈도우의 경우에는 질의 범위 밖에 해당하는 객체를 쉽게 필터링 할 수 있는 반면 원 형태일 경우에는 대부분의 경우 거리 계산을 해야 하는 불편함이 있다.

본 논문에서는 객체의 위치 갱신 연산에 따른 인덱스 구조의 변화를 최소화하기 위해 고정 그리드 방식을 사용하고 VP 필터링 기법을 적용하여 원 범위 질의에서 범위 밖의 객체를 필터링하기 위한 방법을 제시한다.

1. 서론

최근 들어 이동통신 가입자의 급격한 증가로 관련된 서비스가 증가하고 있다. 그중에서도 이동하고 있는 고객의 위치 정보를 반영하여 관련 서비스를 제공하는 LBS(Location Based Service)가 중요한 서비스로 부각되고 있다.

이동객체의 특징은 크게 두 가지로 볼 수 있다. 첫째 연속적으로 이동하기 때문에 빈번한 갱신 연산이 수행된다. 서비스 가입자의 위치변경은 인덱스 구조의 변경을 초래하기 때문에 인덱스 구조가 자주 변하지 않는 구조가 적합하다. 둘째 핸드폰 가입자의 경우 도시의 중심가에 특정 시간에 밀집되는 현상이 발생하는 특징이 있다.

이동 객체를 효율적으로 검색하기 위한 공간 인덱스 구조로 일반적으로 R-트리 구조와 고정 그리드 방식이 대표적인 색인 구조이다. 이동 객체의 경우 빈번한 갱신에 의해 트리 구조를 사용하는 경우 많은 비용을 소요하게 된다. 본 논문에서는 이동 객체의 빈번한 갱신을 저비용으로 처리하기 위한 인덱스 구

조로 고정 그리드 방식을 사용하며 고정 그리드에서 각 셀 간의 순서를 결정하는데 Z-순서(Z-ordering) 기법을 사용한다[1].

특정 시간, 특정 지역에 이동객체가 밀집되어 부하가 집중되는 경우 부하를 분산 시키는 방법으로 비균등 2단계 그리드 구조가 제안되었다[2, 3].

이 구조에서 1단계 인덱스는 그리드내의 각 셀의 부하에 따라 부하 분산을 위해 셀의 모양이 유동적으로 변화하는 단계이며, 2단계 인덱스는 1단계 인덱스의 특정 셀을 다시 고정된 크기로 분할한 그리드 구조이다. 본 논문은 2단계 인덱스에서 고정된 그리드의 범위 질의 처리를 주 대상으로 한다.

주어진 범위 내에서 조건을 만족하는 결과 집합을 찾아내는 범위질의는 많은 LBS 응용에서 중요한 질의 중 하나이다. 예를 들어, 현재 특정 위치로부터 1km 반경 내에 있는 모든 객체 정보를 검색하라 라고 질의 할 수 있다.

인덱스 구조로 Z-순서를 이용한 고정 그리드 방식을 사용할 때, 특정 시간에 특정 지점을 기준으로

윈도우(window)형태나 원(circle) 형태의 범위 질의를 수행하면 Z-순서를 이용하여 범위 내에 있는 셀 집합은 쉽게 구할 수 있지만 셀 집합 내에 있는 모든 객체간의 거리 계산을 수행해야 한다. 매 질의 시마다 해당 셀 내의 모든 객체와의 거리를 계산한다는 것은 비효율적이다. 질의 시간을 줄이기 위해서는 거리 계산을 하지 않고 질의 범위 밖에 있는 객체를 제거 할 수 있는 방법이 필요하다.

본 논문에서는 원 범위 질의시의 거리 계산량을 줄이기 위해 VP 트리와 MVP트리에서 사용된 VP(Vantage Point) 필터링 기법을 적용한다[4, 5]. Z-순서를 이용한 고정 그리드 구조에서 VP 필터링을 이용해 원 범위 질의를 처리하기 위한 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 기술하고, 3장에서 Z-순서를 이용한 고정 그리드 방식에 대해 설명한다. 4장에서는 VP 필터링을 이용한 원 범위 질의 처리에 대해 기술하고, 5장에서 결론과 향후 과제를 기술한다.

2. 관련 연구

2.1 범위 질의의 유형

일반적으로 공간 데이터베이스에서 범위 질의의 경우 윈도우(window) 형태나 원(circle)형태로 이루어진다. 윈도우 형태 질의인 경우에는 윈도우의 왼쪽 하단의 좌표와 오른쪽 상단 좌표만 알면 윈도우의 외곽 부분과 겹치는 셀을 쉽게 구분할 수 있어 그 셀 들만 별도의 거리 계산을 수행하면 된다. 원 형태의 질의인 경우 외곽 부분과 겹치는 셀을 쉽게 찾을 수 없어서 특별한 필터링 방법이 요구된다.

2.2 VP 필터링 기법

VP 필터링 기법의 기본 개념은 공간 도메인 내에 있는 객체들과 질의 점과의 거리 계산시간을 줄이는 것이다. 즉, 질의 중심점과 해당 질의 영역 내에 있는 모든 객체와의 거리 계산을 수행하지 말고 질의 결과가 될 가능성이 없는 객체들은 거리 계산에서 미리 제외하자는 의미이다. 이를 위해서는 어떤 객체가 질의 결과가 될 가능성이 있는지 없는지의 판단을 위한 기준이 필요하다. 이때 기준으로 사용하는 것이 VP(Vantage Point)와 이를 기준으로 하는 각 객체와의 거리 값이다. VP 필터링을 위해서는 VP를 사전에 선정하는 작업과 객체정보가 데

이터베이스 테이블에 삽입 시에 계산되어 반영되어야 한다.

3. Z-순서를 이용한 고정 그리드 방식 인덱싱

고정 그리드(fixed grid) 방식은 주어진 공간 도메인을 일정한 크기의 셀로 분할하여 나누는 방식이다. 그림 1은 이동 객체의 경로 저장을 위해 특정 공간 도메인을 고정 그리드로 분할한 예이다. 특정 객체의 이동 경로는 매 시간 모든 정보가 데이터베이스에 반영되는 것이 아니라 객체가 셀 경계를 벗어날 경우에만 데이터베이스에 반영하게 된다.

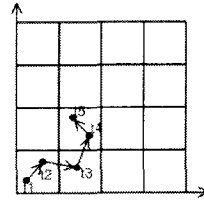


그림 1. 고정 그리드를 이용한 공간 분할

Z-순서는 2D 고정 그리드 인덱스 구조에서 이웃한 셀 들에 순서를 부여하는 여러 방법 중 하나이다. 각 셀의 번호는 기준 좌표로부터 X축, Y축에서 몇 번째 셀인지를 구하여 이진수로 변환한 후 X축, Y축 각 자리의 값을 인터리빙 시켜 구할 수 있다. 특정 객체가 속한 셀 번호를 구하기 위해서는 다음 그림 2와 같은 과정을 거친다.



그림 2. 객체가 포함되는 셀 번호 구하는 과정

그림 3은 Z-순서를 나타내며, 고정 그리드 인덱스 구조에 Z-순서를 사용하면 특정 셀과 일정 거리 내에 인접한 셀을 찾을 때 위에서 설명한 셀 번호 구하는 계산을 역으로 적용하면 쉽게 찾아 낼 수 있다.

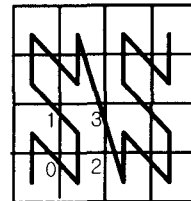


그림 3. Z-순서

그림 4의 테이블은 어떤 객체가 시간 t1에서 t5까

지 이동한 이동 경로를 저장한 테이블 예이다. 위에서 t1, t3, t4는 셀 경계를 벗어나는 이동이므로, 테이블에 저장 되었으며, t2, t5는 셀 내에서의 이동이므로 테이블에 반영되지 않았다.

OID	X	Y	TIME	CELL_NO
O1	10	10	t1	0
O1	140	70	t3	2
O1	160	130	t4	3

그림 4. 이동 객체의 이동경로 저장 예

4. VP 필터링을 이용한 범위 질의 처리

4.1 완전 포함 셀 과 부분 포함 셀

본 논문에서는 VP 필터링을 통해 분류되는 셀을 질의 중심점과 객체간의 별도의 거리계산이 필요 없는 완전 포함 셀과 거리 계산이 필요한 부분 포함 셀로 분류 하였다. 완전 포함 셀과 부분 포함 셀이 되기 위한 조건은 그림 5, 6과 같다.

(조건1) 질의 범위 내에 있는 셀
(조건2) $d(VP, point) < d(VP, cp) - r$ or $d(VP, point) > d(VP, cp) + r$ 인 점을 포함하지 않음

그림 5. 완전 포함 셀 조건

그림 5, 6에서 point는 특정 셀 내에 있는 임의 객체에 대한 좌표이며 cp와 r은 질의 중심점과 반지름이다. $d(VP, point)$ 는 VP와 객체와 거리 이다.

(조건1) 질의 범위 내에 있는 셀
(조건2) $d(VP, point) < d(VP, cp) - r$ or $d(VP, point) > d(VP, cp) + r$ 인 점을 포함

그림 6. 부분 포함 셀 조건

4.2 VP의 결정 및 VP 거리계산 정보 저장

대상 도메인 공간에서 VP를 선택하는 방법은 도메인 공간의 구석에서 한 객체를 선택하는 방법과 경험적 방법 등이 있다. 도메인 공간의 구석에서 VP를 선택하는 방법은 Yianlos에 의해 소개되었다. 실험에 의하면 도메인 공간 내에서 VP를 선택하는

것보다 임의의 한 구석에서 VP를 선택하는 것이 객체의 변별력을 높이는 관점에서 높은 성능을 보이는 것으로 나타나 있다[6]. 경험적 방법은 VP를 선택하기 위해 먼저, 도메인 객체 중에서 임의의 한 객체를 선택하여 다른 모든 객체들과의 거리를 계산한다. 계산된 거리를 기준으로 VP와 가장 먼 거리에 있는 객체를 VP로 선택한다.

본 논문에서는 고정 그리드에서의 질의 특성을 고려하여 양쪽 구석 지점을 VP로 선정하는 방법을 사용하였다. 이 방법은 VP를 선정하기 위해 계산하는 시간을 소비하지 않아도 되며, 검증된 성능을 나타낸다.

그림 7은 결정된 VP1, VP2와 삽입 객체 사이의 거리를 저장하는 컬럼을 추가한 확장된 테이블 구조이다. VP1_DIST와 VP2_DIST는 객체의 위치 정보가 삽입 시에 한번 계산 되어 저장된다.

OID	X	Y	TIME	CELL_NO	VP1_DIST	VP2_DIST

그림 7. VP 거리 컬럼이 추가된 테이블 구조

4.3 원 범위 질의 처리 알고리즘

VP 필터링을 이용한 원 범위 질의 처리 알고리즘은 그림 8과 같다.

```

draw_Circle (cp, r) // cp : central point, r : range
cell_list = Find the cell list in circle using Z-ordering
for each c in overlapped_cell_list
point = some point of c
if exists ((d(VP, point) < (d(VP, cp) - r) OR d(VP, point) > (d(VP, cp) + r)))
insert c into partially_included_cell_list
else if not exists ((d(VP, point) < (d(VP, cp) - r) OR d(VP, point) > (d(VP, cp) + r)))
insert c into fully_included_cell_list
end if
end-for

d = compute distance between cp and each object of partially_included_cell_list
result set = object(c<r) of partially_included_cell_list
result set = result set + all objects of fully_included_cell_list
    
```

그림 8. 원 범위 질의 처리 알고리즘

원 범위 질의 시에 중심점으로부터 연관된 셀 리스트는 해당 중심점을 포함하는 셀 번호를 구한 후 Z-순서 규칙을 이용하여 그 셀과 지정 반경 내에 있는 셀을 구한다.

그림 8의 알고리즘에서 $d(VP, point)$ 는 이동 객체의 이동경로가 저장된 테이블에서 해당 객체의 VP1_DISTANCE와 VP2_DISTANCE를 의미한다.

그림 8의 마지막 부분에서와 같이 VP 필터링을 이용해 완전포함 셀과 부분 포함 셀을 구분해 낸 후 완전 포함 셀 내의 모든 객체는 질의 결과에 모두 포함되며, 부분 포함 셀에 해당하는 객체는 중심점과의 거리계산을 다시 수행한 후에 결과 내에 포함할지를 결정해야 한다.

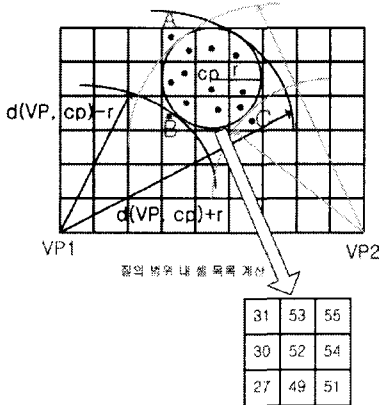


그림 9. 원 범위 질의 처리 예

그림 9에서 원 범위 질의 중심점 cp로 부터 범위 r안에 있는 셀 집합을 Z-순서를 이용해 구해보면 9개의 셀을 구할 수 있다. 예를 들어, 각 셀 번호가 그림 9의 밑에 그림과 같다고 한다면 30, 49, 52, 53, 54번 5개의 셀은 완전 포함 셀 이므로 별도의 계산 없이 5개의 셀 안에 포함된 모든 객체가 질의 결과에 포함된다. 27, 31, 51, 55번 에 해당하는 4개의 셀 중에서 55번 셀은 VP1에 대해 완전 포함 셀의 조건을 만족하므로 마찬가지로 별도의 계산 없이 모든 객체가 결과 집합에 포함된다.

27은 VP1에 대해서 부분 포함 셀 조건을 만족하는 객체 B를 포함하므로 부분 포함 셀에 해당되며 마찬가지로 31, 51 셀도 VP2에 대해서 A, C가 부분 포함 셀 조건에 해당 되므로 부분 포함 셀이 된다. 결과적으로 27, 31, 51셀 내의 객체에 대해서만 거리 계산을 다시 수행하면 된다.

5. 결론

위치 기반 서비스에서 연속적으로 저장하는 이동 객체를 인덱스 구조의 변경을 최소화 하며 위치 저장 하는 방법과 효율적인 범위 질의는 중요 부분으로 부각되고 있다 .

본 논문에서는 이동객체의 위치 변경에 따른 인덱스 갱신 연산을 최소화하기 위해 고정 그리드 방식을

사용하였다. 또한 효과적인 원 범위 질의 처리를 위해 VP 트리, MVP 트리에서 사용하는 VP 필터링 기법을 적용하고, VP 필터링을 이용한 질의 처리를 위한 알고리즘을 제안하였다.

향후 연구과제로는 제안된 방법에 대한 성능평가와 고정 그리드 인덱스 방식에서 최근접 질의를 위한 VP 필터링 기법이 연구되어야 한다.

참고문헌

- [1] 박원순, 전세길, 나연목, "대용량 이동 객체의 위치 정보 인덱싱", 정보과학회 추계학술발표 논문집, Vol. 29, No. 2, 2002년 10월, pp 49-51.
- [2] Y.Nah, M.H.Kim, T.Wang, K.H.Kim, Y.K.Yang, "TMO-structured Cluster-based Real-time Management of Location Data on Massive Volume of Moving Items," STFES 2003, Hakodate, Japan.(to be published)
- [3] M.H.Kim, T.Wang, K.H.Kim, Y.Nah, Y.K.Yang, "Distributed Adaptive Architecture for Managing Very Large Volume of Moving Items," IDPT 2003, Beijing, China. (to be published)
- [4] Jeffrey K. Ujlmann, "Satisfying General Proximity/Similarity Queries with Metric Trees, " Information Processing Letters, Vol. 40, pages 175-179. 1991.
- [5] Tolga Bozkaya and Meral Ozsoyoglu, "Distance-Based Indexing for High-Dimensional Metric Spaces," Proceedings of the ACM SIGMOD Conference, pages 357-368. 1997.
- [6] P.N.Yiannilos, "Data Structures and Algorithms for Nearest Neighbor search in General Metric Spaces," ACM-SIAM Symposium on Discrete Algorithms, 1993, pages 311-321.
- [7] 김병곤, 이재호, 임해철, "R-tree 계열의 인덱싱 구조에서의 효율적 질의 처리를 위한 VP 필터링," 한국정보과학회 논문지, Vol. 29 No. 6, 2002년 12월, pp. 453-463.
- [8] Philippe Rigaux, Michel, Scholl, Agnes Voisard, *Spatial Databases with Application to GIS*, Morgan Kaufmann.