

## 위치 기반 서비스에서 연속 범위 질의와 k-최근접 질의 처리에 대한 조사

A Survey on Continuous Range and k-Nearest Neighbor Query Processing in Location Based Services

---

저자 (Authors)	이성민, 정하림, 정연돈, 이기용 Sungmin Yi, HaRim Jung, Yon Dohn Chung, Ki Yong Lee
출처 (Source)	<a href="#">정보과학회논문지 : 데이터베이스 38(1)</a> , 2011.2, 49-63 (15 pages) <a href="#">Journal of KISS : Databases 38(1)</a> , 2011.2, 49-63 (15 pages)
발행처 (Publisher)	<a href="#">한국정보과학회</a> KOREA INFORMATION SCIENCE SOCIETY
URL	<a href="http://www.dbpia.co.kr/Article/NODE01601916">http://www.dbpia.co.kr/Article/NODE01601916</a>
APA Style	이성민, 정하림, 정연돈, 이기용 (2011). 위치 기반 서비스에서 연속 범위 질의와 k-최근접 질의 처리에 대한 조사. 정보과학회논문지 : 데이터베이스, 38(1), 49-63.
이용정보 (Accessed)	성균관대학교 과학학술정보관 115.***.170.150 2017/09/06 15:14 (KST)

---

### 저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

### Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

# 위치 기반 서비스에서 연속 범위 질의와 k-최근접 질의 처리에 대한 조사

## (A Survey on Continuous Range and k-Nearest Neighbor Query Processing in Location Based Services)

이 성 민 <sup>\*</sup>      정 하 림 <sup>\*</sup>      정 연 돈 <sup>\*\*</sup>      이 기 용 <sup>\*\*\*</sup>  
(Sungmin Yi)      (HaRim Jung)      (Yon Dohn Chung)      (Ki Yong Lee)

**요 약** 최근 위치기반 정보 서비스는 GPS를 탑재한 스마트폰의 대중화 및 다양한 무선통신 기술의 발달에 힘입어 새로운 성장 국면을 맞이하고 있다. 본 논문에서는 효과적인 위치기반 정보 서비스 제공의 초석이 될 수 있는 위치 기반 질의 처리 기법 동향을 소개·분석한다 (본 논문에서 소개하는 연속 공간 범위 질의 기법들은 VCI, Q-Indexing, MQM, MobiEyes, P2P-MQM, 연속 k-최근접 질의 기법들은 Yu-CNN, SEA-CNN, CPM, iSEE, disMKNN, P2P-MQM이다). 특히, 질의 및 데이터의 위치 변화에 따라 지속적으로 질의 결과의 유효성을 검증하고 새로운 결과를 신속하고 에너지 효율적으로 제공하기 위한 연속 위치기반 질의 처리 기법에 초점을 맞추어 진행한다. 또한, 기존의 위치기반 질의의 한계점과 향후 연구 방향을 제시한다.

**키워드** : 위치 기반 정보 서비스, 위치 기반 질의, 연속 질의, 범위 질의, k-최근접 질의

**Abstract** With the wide spread of smart phones and the development of wireless communication technologies, the popularity of location-based information services has recently gained momentum. In this paper, we study on the methods of processing location-based queries, which are one of the core building blocks for supporting location based information services (This paper covers the following methods - VCI, Q-Indexing, MQM, MobiEyes, and P2P-MQM for continuous spatial range queries, and Yu-CNN, SEA-CNN, CPM, iSEE, disMKNN, and P2P-MQM for continuous k-NN queries). In particular, we focus on the methods for fast and energy-efficient processing of continuous location-based queries, which require continual re-evaluation as their results become easily invalid due to the movement of queries and/or data. In addition, we present a new research direction inspired by the limitations of existing work.

**Key words** : Location-based information services, Location-based queries, Continuous query, Range query, k-nearest neighbor query

· 본 연구는 삼성전자의 연구비 지원으로 수행되었음

<sup>\*</sup> 학생회원 : 고려대학교 컴퓨터전파통신공학과  
winysm@korea.ac.kr  
harim3826@korea.ac.kr

<sup>\*\*</sup> 종신회원 : 고려대학교 컴퓨터·통신공학부 교수  
yidchung@korea.ac.kr

<sup>\*\*\*</sup> 정 회 원 : 숙명여자대학교 컴퓨터과학과 교수  
kiyonglee@sookmyung.ac.kr

논문접수 : 2010년 8월 18일

심사완료 : 2010년 10월 6일

Copyright©2011 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: 데이터베이스 제38권 제1호(2011.2)

## 1. 서 론

최근 GPS를 탑재한 스마트폰 보급의 확산 및 다양한 무선통신 기술의 발전에 힘입어 시간·장소의 제약 없이 다양한 정보 서비스를 제공받을 수 있는 시대가 가시화되고 있으며, 사용자의 위치 정보를 활용하여 사용자에게 더욱 가치 있는 정보를 제공하는 위치기반 정보 서비스(location-based information services)에 대한 관심이 국내·외적으로 크게 증가하는 추세이다. 디지코<sup>1)</sup>의 연구 리포트 “아이폰 고객의 아이폰/IT서비스 이용 형태 및 시사점”(2010.02.04)에 따르면 국내 아이폰 고객 1,400명을 대상으로 한 조사에서 다양한 애플리케이션

1) <http://www.digieco.co.kr>

선 중 위치기반 정보 서비스는 전 연령층에서 고른 분포로 활용빈도가 높게 나타났다.

위치기반 정보 서비스를 효과적으로 지원하기 위해서는 사용자의 현재 위치를 기반으로 하여 정보/데이터를 탐색하는 위치기반 질의(location-based queries)의 효율적인 수행이 필수적이다.

범위 질의(range queries)와 k-최근접 질의(k-nearest neighbor queries: k-NN queries)[1-20]는 다양한 위치기반 질의 중 가장 기본적인 질의이다. 여기서 공간 범위 질의란 “현재 위치에서 반경 3km 이내에 위치하는 모든 음식점을 검색하라.”와 같이 주어진 범위 안의 모든 데이터들을 검색하며, k-최근접 질의는 “현재 위치에서 가장 가까운 k개의 음식점을 검색하라.”와 같이 사용자의 현재 위치로부터 가장 가까운  $k(\geq 1)$ 개의 데이터들을 검색한다. 이외에도 역 k-최근접 질의(reverse k-NN queries)[3], 그룹 최근접 질의(group NN queries)[4], 범위 k-최근접 질의(range k-NN queries)[5] 등 공간 범위 질의/k-최근접 질의의 다양한 변형들이 존재하지만 본 논문에서는 다루지 않는다.

위치기반 질의는 질의 및 데이터의 이동성에 따라 (1) 고정(static) 데이터에 대한 일회성(snapshot) 질의, (2) 고정 데이터에 대한 이동(moving) 질의, (3) 이동 데이터에 대한 고정 질의, (4) 이동 데이터에 대한 이동 질의와 같이 4가지로 분류할 수 있다. 특히 2, 3, 4 번 유형의 질의는 질의 또는 데이터가 이동함에 따라 질의의 결과가 끊임없이 변하므로 지속적으로 모니터링(monitoring)해야 하며, 이러한 유형의 질의를 연속 위치기반 질의(continuous location based queries)라고 한다. 연속 질의는 주행하고 있는 자동차에게 지속적으로 가장 가까운 주유소 정보를 제공해 주는 gas station finder 서비스(유형 2), 점심시간 동안 음식점 반경 3km 이내에 있는 고객들에게 할인 쿠폰을 발송하는 e-coupon 서비스(유형 3), 주행 중인 택시에게 택시를 찾고 있는 가장 가까운 승객의 위치를 제공해 주는 cab finder 서비스(유형 4) 등과 같이 다양한 위치기반 정보 서비스에 적용될 수 있다.

연속 위치기반 질의는 오랜 시간 동안 지속적으로 수행되므로 질의 및 데이터의 이동성을 고려하여 지속적으로 결과를 갱신해야 한다. 이러한 연속 위치기반 질의를 효과적으로 처리하기 위하여 제안되었던 기존 기법들은 질의 및 데이터의 잦은 위치 변화를 극복하기 위해 메인 메모리 기반 색인 구조를 활용하여 연산을 수행하며, 입·출력 오버헤드를 줄이기 위한 일회성 위치기반 질의 처리 기법들과는 달리 서버의 CPU 오버헤드 및 서버-이동 데이터 혹은 서버-이동 사용자 사이의 통신 오버헤드 절감을 목표로 한다.

본 논문에서는 지금까지 제안되었던 위치기반 질의(범위 질의와 k-최근접 질의) 처리 기법들 중 연속 위치기반 질의 처리 기법들에 더욱 초점을 맞추어 비교·분석하고 향후 연구 방향을 제시한다.

## 2. 범위 질의 처리 기법

### 2.1 고정 데이터에 대한 일회성 범위 질의 처리 기법

고정 데이터에 대한 일회성 공간 범위 질의의 효율적인 처리를 위해 다양한 공간 혹은 다차원 색인 구조들이 제안되었다. 그 중, R-tree 및 이의 변형 색인 구조들은 그 효율성으로 인해 IBM Informix, Oracle, MySQL과 같은 상업적 데이터베이스 시스템뿐만 아니라, PostgreSQL, SQLite와 같은 오픈 소스 데이터베이스 시스템에서도 공간 데이터를 색인 하는데 널리 사용되고 있다[1,8,9].

그림 1과 같이 R-tree는 공간적으로 서로 인접한 데이터들의 위치를 MBR(minimum bounding rectangle)로 근사화(approximation)하여 계층화된 트리 구조를 구축하므로, 범위 질의 처리에 매우 효과적이다. R-tree에서 범위 질의 처리는 루트를 시작으로 주어진 질의 범위와 겹치는 노드만을 접근·검사한다. 예를 들어, 그림 1에서 질의 범위  $q.R$ 이 주어졌을 때, 루트노드를 시작으로  $q.R$ 과 겹치는 R-tree 노드인  $N_2$ ,  $N_6$ 를 접근하여 데이터  $d_7$ 를 결과 값으로 반환한다. 디스크 기반

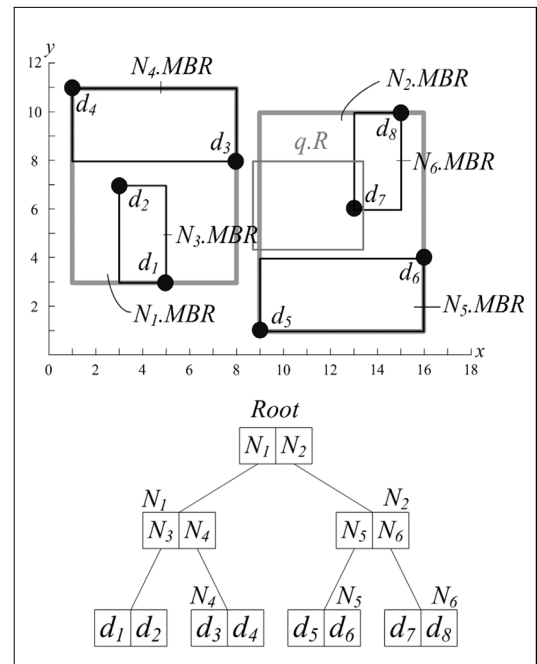


그림 1 R-tree의 예

R-tree는 고정 데이터에 대한 일회성 범위 질의 처리 시 좋은 성능을 보이지만, 잦은 갱신 연산(e.g., 삽입/삭제 연산)을 초래하는 이동 데이터에 대한 연속 범위 질의의 처리를 효과적으로 지원하지 못한다.

## 2.2 연속 범위 질의의 처리 기법

연속 범위 질의를 처리하기 위한 가장 단순한 방법은 이동 데이터 및 질의(클라이언트)의 위치가 변할 때마다 서버에게 자신의 위치를 보고하게 하는 것이다. 그 후, 서버는 데이터의 위치를 기반으로 메인메모리 혹은 디스크 기반 색인 구조를 구축하고 지속적으로 색인 구조를 갱신하고 질의를 처리한다. 하지만, 수많은 이동 데이터들이 지속적으로 보고하는 대량의 위치정보에 대한 색인 구조의 빈번한 갱신 및 많은 수의 연속 범위 질의의 처리는 서버의 병목현상을 초래할 수 있다. 또한, 이동 데이터 및 질의의 지속적인 위치보고는 많은 양의 에너지를 소모하게 된다. 본 절에서는 이러한 문제를 극복하기 위한 주요 연구들을 소개한다.

### 2.2.1 VCI

연속 범위 질의를 처리하기 위한 R-tree의 변형으로 TPR-tree(time parameterized R-tree)[11,12]가 대표적이며, 이동 데이터들의 특정 시각에서의 위치와 속도 벡터 정보를 이용하여 이들의 위치를 표현하고 저장하므로 이동 데이터들의 현재 및 미래 위치의 예측을 통해 연속 범위 질의를 처리할 수 있다. 하지만, TPR-tree는 데이터의 고정된 속도 및 방향을 가정하여 색인하므로, 데이터가 일정한 속도 및 방향 패턴을 가지지 않는 실제 환경에 적용하기 힘들다. 한편, Velocity Constrained Index(VCI)[13]가 있다. 특히, VCI는 데이터의 이동에 의한 색인 구조의 빈번한 갱신 연산을 줄이기 위해 제안되었으며, 각 데이터의 최대 속도를 활용한다. VCI의 노드들은 자신의 MBR이 포함하고 있는 데이터들의 최대 속도 중 최대값인  $V_{max}$ 를 추가적으로 저장하여 색인한다. 특정 시점  $t_0$ 에서 VCI를 구축하였다고 가정하자. 이 시점에서 VCI의 MBR들은 모든 이동 데이터들의 위치를 정확하게 반영한다. 이때, 특정 미래 시점  $t_x (> t_0)$ 에서 연속 범위 질의의 처리는 각각의 MBR을  $V_{max}$ 를 활용하여 각 방향마다  $R = V_{max} \times (t_x - t_0)$ 만큼 확장 후 처리할 수 있다. 그림 2는 이러한 MBR의 확장을 보인다.

VCI는 고정된 속도 및 방향을 가정하는 TPR-tree와는 다르게 "데이터들이 특정 속도( $V_{max}$ )보다 빠르지 않다."라고만 가정함으로써 이동 방향 및 속도가 끊임없이 변하는 데이터를 가정하는 실제 응용에 더 유용하게 사용될 수 있다. 하지만, 지속적인 MBR의 확장은 결과가 아닌 데이터들을 결과에 포함한다고 간주하는 경우가 많이 발생할 수 있으므로 실제 결과를 위한 필터링 단

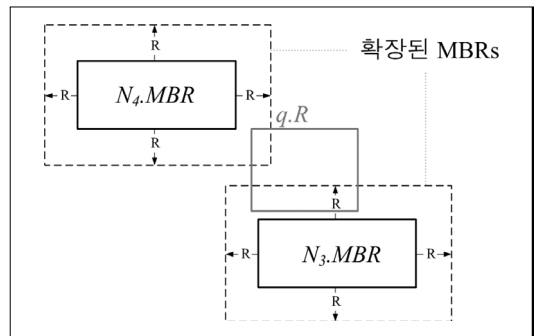


그림 2 VCI에서 MBR 확장의 예

계가 필요하며, 특정 주기마다 각 이동 데이터의 현재 위치 및 최대 속도를 기반으로 VCI를 재구축해야 한다. 이처럼 특정 주기에만 이동 데이터가 자신의 위치를 서버에게 보고함으로써 서버-이동 데이터 사이의 통신비용 및 색인 구조의 갱신 연산을 줄일 수 있다.

### 2.2.2 Q-indexing

[13]에서는 예측할 수 없는 속도와 방향으로 이동하는 데이터들을 색인 하였을 때, 질의 처리 성능 감소와 직접적으로 관련되는 색인 구조의 빈번한 갱신을 줄이기 위해서 데이터를 색인하는 것이 아닌 질의를 색인하는 질의 색인(Query indexing: Q-indexing) 패러다임이 제안되었다. 특히, 질의 색인은 이동 데이터에 대한 연속 고정 범위 질의 처리(유형 3)에 적합하다. 그 이유는 질의가 이동하지 않으면서 장시간 수행되기 때문이다.

그림 3은 질의  $q_1, q_2, \dots, q_8$ 를 색인하는 디스크 기반 R-tree를 보인다. 예를 들어, 이동 데이터  $d_7$ 가 그림 3과 같이 이동하였을 경우, 루트 노드를 시작으로  $d_7$ 를 포함하는 R-tree 노드인  $N_2, N_6$ 를 접근하여  $d_7$ 를 포함하는 질의  $q_7$ 를 찾은 후  $q_7$ 의 결과에  $d_7$ 를 포함시킨다.

또한, [13]에서는 서버의 효율적인 질의 처리를 위해, 각각의 이동 데이터들에게 안전 영역(safe region)을 부여함으로써 색인 구조의 갱신 비용을 줄였다. 안전 영역은 그림 4와 같이 이동 데이터의 위치변화가 안전 영역 내부에서 이루어 질 경우 어떠한 질의의 결과에도 영향을 미치지 않는 영역이다. 따라서 이동 데이터는 서버에게 자신의 위치 갱신 메시지를 전송할 필요가 없으므로, 서버-이동 데이터 사이의 통신비용을 줄일 수 있다. 하지만 서버가 각 이동 데이터의 안전 영역을 부여하기 위해 많은 연산을 소모하게 되므로 실시간으로 대량의 질의를 처리해야 하는 실제 응용에 적용하기 힘들다. 한편, [14]에서는 실시간 LBS 응용을 위한 질의 처리 시간 단축을 위해서 디스크 기반이 아닌 메인 메모리 기반 그리드 색인(grid index) 구조를 사용하여 질의를 색인하여 연속 범위 질의를 처리하였다.

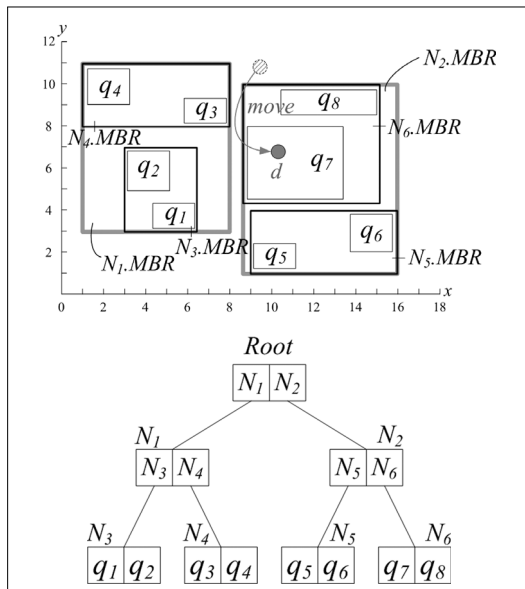


그림 3 질의를 색인하는 R-tree의 예

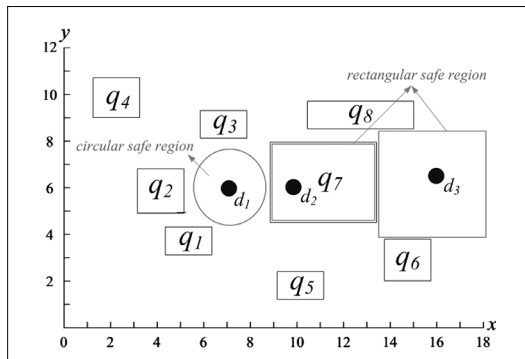


그림 4 안전 영역(safe region)의 예

### 2.2.3 MQM

MQM(Monitoring Query Management)[15]은 기존의 서버 중심의 연산 환경이 아닌 다양화된 휴대용 단말기(이동 데이터)의 연산 자원 능력을 바탕으로 분산화된 연산 환경에서 연속 고정 범위 질의를 처리하고자 하였다. 여기서 이동 데이터의 연산 자원 능력이란 해당 이동 데이터가 자신과 인접한 질의들을 할당 받아 지속적으로 처리하고 관리할 수 있는 능력을 말한다.

특히, MQM은 상주 영역(resident domain) 개념을 도입하였다. 상주 영역이란 각각의 이동 데이터가 직접 연산해야 하는 질의를 포함하는 영역이며, 이러한 영역은 이동 데이터의 연산 능력에 따라 결정되고 할당된다. 그림 5는 이러한 상주 영역의 예를 보여준다. 이동 데이터는 서버에 상주 영역을 할당받고 해당 상주 영역 내

의 모든 질의를 인식하여 자신의 위치가 변할 때 마다 다음 2가지의 상황을 체크한다:

- 상주 영역 밖으로 이동하는 경우: 만약 이동 데이터가 상주 영역 밖으로 이동하였다면(그림 5의 case 1) 자신의 현재 위치와 함께 서버에게 새로운 상주 영역을 요청한다.
- 상주 영역에 포함된 질의를 가로지르는 경우: 만약 이동 데이터가 상주 영역에 포함된 질의의 범위 내로 이동 혹은 범위 밖으로 이동하였다면(그림 5의 case 2) 서버에게 자신의 현재 위치와 함께 해당 질의 결과의 갱신을 요청한다.

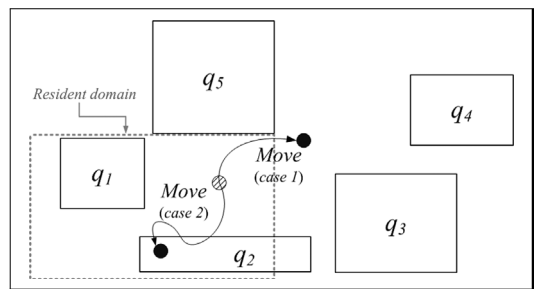


그림 5 상주 영역(resident domain)의 예

이처럼 각각의 이동 데이터는 자신의 상주 영역 내의 모든 질의를 인식하고, 질의 연산을 직접 수행함으로써 기존 서버 중심의 연산 환경의 문제점인 서버의 병목현상을 완화시켰다. 또한 위의 2가지 상황의 경우에만 서버에 요청(새로운 상주 영역의 요청, 질의 결과 갱신)하므로 연속 범위 질의 처리를 위한 서버-이동 데이터 사이의 통신비용을 줄였다.

MQM에서는 각각의 이동 데이터에 대한 상주 영역을 결정하고 할당하기 위해 메인 메모리 기반 색인 구조인 Binary Partitioning tree(BP-tree)가 사용된다. BP-tree는 전체 도메인  $D$ 를 시스템에 등록 된 이동 데이터들의 연산 자원(처리할 수 있는 질의의 개수) 중 최소를 분할 임계값(split threshold)  $\theta$ 로 정하고 재귀적으로  $D$ 를 이진 분할하여 트리 구조를 구축한다.

그림 6은 분할 임계값이 2일 때 BP-tree의 예를 보인다. BP-tree는 비단말 노드(non-leaf node)인 도메인 노드(domain node)와 단말 노드(leaf node)인 데이터 노드(data node)로 구성된다. 도메인 노드  $D$ 는 두 개의 엔트리( $R, P$ )를 포함한다. 여기서  $R$ 은  $D$ 의 서브도메인으로서 포인터  $P$ 가 가리키는 자식 노드의 도메인이다. 데이터 노드는 자신의 도메인에 포함되는 혹은 겹치는  $\theta$ 개 이하의 질의들을 유지한다. 각각의 질의는 1개 이상의 데이터 노드에 삽입될 수 있다. 예를 들어, 질의

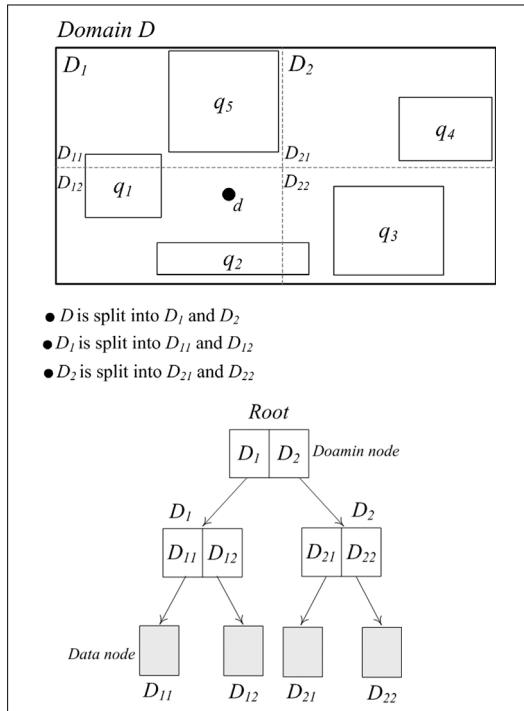


그림 6 BP-tree의 예

$q_2$ 는 데이터 노드  $D_{12}$ ,  $D_{22}$ 의 도메인에 겹치므로  $D_{12}$ ,  $D_{22}$ 에 분할되어 삽입된다. 모든 도메인 노드와 데이터 노드는 추가적으로 자신의 도메인에 겹치는 질의들의 개수의 총합인 size를 포함한다( $D_1.size = 4$ ,  $D_{11}.size = 2$ ,  $D_{12}.size = 2$ ). 그림 6에서 이동 데이터  $d$ 의 연산 능력을 2라고 했을 때,  $d$ 의 상주 영역은  $d$ 를 포함하며  $D.size$ 가 2보다 작거나 같은 노드  $D_{12}$ 의 도메인이 되고, 서버는 해당 도메인에 겹치는 모든 질의들  $q_1$ ,  $q_2$ 를 상주 영역과 함께  $d$ 에게 할당한다.

#### 2.2.4 MbiEyes

MobiEyes[16]는 MQM과 마찬가지로 서버 연산의 병목현상, 서버와 이동 데이터 사이의 통신비용을 줄이기 위해 이동 데이터의 연산 자원 능력을 활용한 분산화된 연산 환경을 가정한다. 하지만 MQM과는 다르게 이동 데이터에 대한 연속 이동 범위 질의를 처리한다. 이를 위하여 MobiEyes는 이동 데이터 및 이동 범위 질의는 자신의 속도 및 방향을 알 수 있다고 가정하며 이를 활용한다.

MobiEyes는 그리드 색인 구조를 사용한다. 각 이동 범위 질의의 질의 범위는 원으로 가정하며 경계 사각형(bounding box)로 표현한다. 여기서 경계 사각형은 질의 범위  $q.R$ 의 중점  $q.P$ 가 특정 그리드 셀 안에서만 이동할 경우  $q.R$ 이 이동할 수 있는 모든 영역을 포함하는

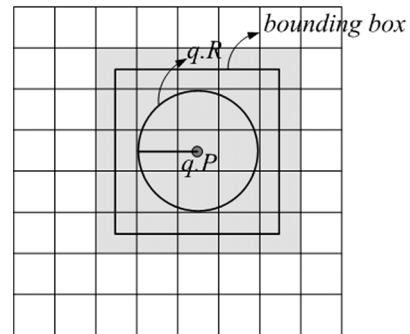


그림 7 모니터링 영역(monitored region)의 예

사각형이다(그림 7 참조). MobiEyes는 각각의 이동 데이터가 직접 연산해야 하는 질의를 할당하기 위해 모니터링 영역(monitored region) 개념을 활용한다. 여기서 모니터링 영역이란 질의의 경계 사각형과 교차되는 셀들의 집합이다(그림 7의 음영 처리된 영역).

이동 데이터는 자신의 위치가 특정 이동 질의의 모니터링 영역에 포함되어 있다면 그 질의를 속도 벡터와 함께 서버에게 할당 받고 주기적으로 질의 연산을 수행한다. MQM과 마찬가지로 자신이 질의의 범위 내로 이동 혹은 범위 밖으로 이동하였을 경우에만 서버에게 해당 질의 결과의 갱신을 요청한다. 이동 질의는 (1) 속도 벡터가 크게 변했을 경우와 (2) 질의 범위의 중점이 현재 그리드 셀에서 다른 셀로 이동하였을 경우에만 서버에게 그 사실을 보고 한다. 이와 같은 메커니즘을 통해 MobiEyes는 연속 범위 질의 처리를 위한 서버의 병목현상 및 서버-이동 데이터/서버-이동 질의 사이의 통신비용을 줄였다.

#### 2.2.5 P2P-MQM

P2P-MQM[21]은 서버가 존재 하지 않는 모바일 애드혹(mobile ad-hoc) 환경/모바일 P2P 환경을 가정하며, 그리드 기법을 사용한다. 그리드의 각 셀은 이동 데이터의 통신 능력에 기인하여  $\frac{r}{\sqrt{2}}$  크기의 정사각형으로 설정한다. 여기서,  $r$ 은 전송 영역(transmission range)의 반지름이다. 각각의 이동 데이터는 자신이 속해 있는 셀 포함되는 혹은 겹치는 질의들의 정보를 저장하며 자신의 위치 변화가 질의의 결과에 영향을 주게 될 경우, 질의 생성자(query creator)에게 질의 결과의 갱신을 요청한다. 만약, 각 이동 데이터  $d$ 가 이전 셀  $c$ 에서 새로운 셀  $c'$ 으로 이동할 경우, 데이터는  $c'$ 에 존재하는 데이터들에게  $c'$ 에 포함되는 혹은 겹치는 질의들을 요청하여 연속 범위 질의를 수행한다. 한편, 이전 셀  $c$ 에 이동 데이터가 존재하지 않을 경우,  $d$ 는  $c$ 의 질의 정보를 저장하며,  $c$ 에 새로운 이동 데이터가 들어오면

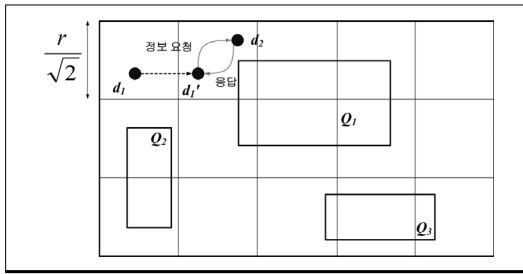


그림 8 P2P-MQM의 예

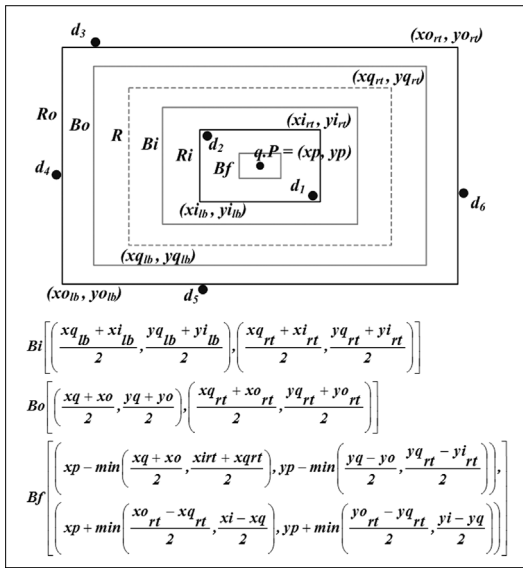


그림 9 가상의 사각형 Ri, R, Ro, Bf, Bi, Bo

이를 제공하고 삭제한다. 만약  $d_i$ 가  $c'$ 보다  $c$ 에서 더 먼 셀  $c'$ 로 이동할 경우,  $c'$ 에 존재하는 다른 이동 데이터에  $c$ 의 질의 정보를 저장하도록 한다. 그림 8은 P2P-MQM의 예이다.

이동 데이터에 대한 연속 이동 범위 질의를 처리하기 위해 P2P-MQM은 그림 9와 같이 6개의 범위  $Ri$ ,  $R$ ,  $Ro$ ,  $Bf$ ,  $Bi$ ,  $Bo$ 를 정의한다. 전체 이동 데이터의 집합을  $D = \{d_1, d_2, \dots, d_{|D|}\}$ 라고 하자.

이 때,  $Ri$ 는 질의 범위 안에 포함되는 이동 데이터들의 집합  $D' (\subset D)$ 를 포함하는 최소 범위,  $Ro$ 는  $D'$ 를 포함하는 최대 범위,  $R$ 은 질의 범위,  $Bf$ 는 자신의 내부에서 질의 생성자가 이동하여도 질의 결과의 변화가 없음을 보장하는 범위,  $Bi$ 는 자신의 내부에서  $R$ 에 포함되는 각 데이터  $d' (\in D')$ 가 이동 하여도 질의 결과의 변화가 없음을 보장하는 범위,  $Bo$ 는  $R$ 의 외부에 존재하는 각 데이터  $d (\in D - D')$ 가 자신의 내부로 이동하지 않으면 질의 결과가 변화하지 않음을 보장하는 범위가

다. 이러한 범위들을 이용하여 질의 생성자가  $Bf$ 를 벗어 날 때, 질의 범위  $R$  내의 데이터가  $Bi$ 를 벗어 날 때,  $R$ 의 외부에 존재하는 데이터가  $Bo$  내부로 이동할 때 연속 이동 질의의 갱신 연산을 수행한다.

P2P-MQM은 이동 데이터에 대한 연속 고정 범위 질의뿐만 아니라 이동 데이터에 대한 연속 이동 범위 질의, 이동 데이터에 대한 연속 고정  $k$ -최근접 질의, 이동 데이터에 대한 연속 이동  $k$ -최근접 질의 등도 처리할 수 있다.  $k$ -최근접 질의 관련 내용은 3장에서 추후에 설명 한다.

### 2.3 비교분석

2장에서 연속 범위 질의 처리를 위한 연구 VCI [13], Q-Indexing [13], MQM [15], MobiEyes [16], P2P-MQM [21]을 소개하였다. VCI는 이동 데이터의 최대 속도를 추가적으로 저장한 디스크 기반 R-tree의 변형 색인 구조를 이용하여 특정 주기에만 이동 데이터가 자신의 위치를 서버에게 보고하게 함으로써 서버-이동 데이터 사이의 통신비용 및 색인 구조의 갱신 연산을 줄일 수 있다. 그러나 디스크 기반 R-tree의 사용으로 메인 메모리 기반 색인구조보다 검색시간이 오래 걸린다는 단점이 있다. Q-Indexing은 색인 구조의 갱신 연산 비용을 줄일 수 있도록 디스크 기반 R-tree 혹은 메인 메모리 기반 그리드를 사용하여 질의를 색인함으로써 색인 구조의 빈번한 갱신을 줄였다. 또한, 서버와 이동 데이터 사이의 통신비용을 줄이기 위하여 안전 영역 개념을 도입하였으나, 안전 영역의 대부분은 사이즈가 작으며 안전 영역의 재설정 비용이 크다는 단점이 존재한다 (안전 영역의 설정은 질의 개수  $n$ 에 따라서  $O(n)$ 에서  $O(n \log^3 n)$ 까지 비용이 소모된다).

MQM은 기존의 서버 중심의 연산 환경이 아닌 이동 데이터의 연산 자원 능력을 활용한다. 이를 위해 MQM은 상주 영역 개념을 도입하였으며 각 이동 데이터에게 상주 영역을 할당하기 위한 BP-tree를 소개하였다. 이동 데이터는 새로운 상주 영역의 요청 및 질의 결과 갱신 요청을 위한 메시지만을 송신하므로 서버와 이동 데이터 사이의 통신비용을 줄일 수 있다. 그러나 데이터의 상주영역에 따라 질의를 분할하여 BP-tree를 구축하므로 서버에 등록된 질의의 개수보다 더 많은 개수의 질의 범위를 관리해야 하므로 이동 데이터의 연산 자원 능력을 충분히 활용하지 못하는 단점이 존재한다. MobiEyes는 이동 데이터 및 이동 질의를 속도 벡터를 활용하며 메인 메모리 기반 그리드 색인 구조를 사용한다. 또한, 이동 데이터의 연산 자원 능력을 활용하기 위하여 모니터링 영역 개념을 도입하여 서버의 질의 처리 비용을 줄였으며, 이동 데이터가 자신의 속도 및 방향이 크게 변화하였을 경우 및 질의 결과 갱신 요청을 위한 메시지

만을 송신하므로 서버와 이동 데이터 사이의 통신비용 및 서버의 병목현상을 줄였다. 그러나 수시로 이동 데이터의 속도와 방향이 변화하는 동적 환경에서는 정확한 결과를 도출해 낼 수 없다. P2P-MQM은 중앙서버가 존재하지 않는 모바일 애드혹 혹은 모바일 P2P 환경에서 다양한 위치기반 질의 유형 및 연속 질의 유형을 지원하고 서버 구축이 힘든 환경에서도 연속 질의 처리를 지원할 수 있다는 장점이 있지만, 서버의 부재로 인해 이동 데이터의 연산 능력이 시스템 성능을 크게 좌우하게 된다. 표 1은 위에서 소개한 연속 범위 질의 처리를 위한 연구 내용에 대한 요약 및 비교 분석을 보이고 있다. 또한, 각 기법의 성능 평가는 해당 기법의 논문들을 참조하길 바란다.

### 3. k-최근접 질의 처리 기법

#### 3.1 고정 데이터에 대한 일회성 k-최근접 질의 처리 기법

본 논문에서는 고정 데이터에 대한 일회성 k-최근접 질의 처리도 범위 질의 처리와 마찬가지로 R-tree에 초점을 맞춘다. R-tree 상에서 k-최근접 질의의 효율적인 처리에 초점을 맞춘 많은 알고리즘들이 제안되었다 [2,6,7]. 이들은 [2]에서 제안한 거리척도인 *mindist*를 활용한 분기한정(branch-and-bound) 탐색 방식을 따른다. 질의 점  $q.P$ 와 R-tree 노드  $N$ 이 주어졌을 때,  $mindist(q.P, N.MBR)$ 는  $N.MBR$ 이 이 거리보다 가까운 데이터를 포함하지 않음을 보장한다. 그림 10에서는 간략함을 위해  $mindist(q.P, N_b.MBR)$ 만을 보이고 있다.

[7]에서 제안한 최적 우선(best-first) 탐색 알고리즘은 R-tree에서 k-최근접 질의 처리에 가장 효율적인 방법으로 알려져 있다. 최적 우선 탐색 알고리즘은 *mindist*를 키 값으로 하여 오름차순으로 정렬하는 힙(heap)을 이용한다. 그림 8은 질의 점  $q.P$ 에서의 2-최근접 질의를 최적 우선 탐색 알고리즘을 사용하여 검색하는 예

표 1 연속 범위 질의 처리 주요 연구

주요 연구	연속 질의 유형			주요 내용	장점	단점
	고정 데이터 / 이동 질의	이동 데이터 / 고정 질의	이동 데이터 / 이동 질의			
VCI [13]	×	○	×	· 이동 데이터의 최대 속도를 추가적으로 저장한 디스크 기반의 R-tree 변형 색인 구조	· 서버-이동 데이터 사이의 통신비용 및 색인 구조의 갱신연산 비용을 줄임	· 메인 메모리 기반 색인구조보다 검색시간이 오래 걸림
Q-Indexing [13,14]	×	○	×	· 고정 연속 질의를 디스크 기반 R-tree 혹은 메인 메모리 기반 그리드를 사용하여 색인 · 안전 영역(safe region) 개념을 도입	· 질의 색인을 통하여 색인 구조의 갱신 연산 비용을 줄임 · 안전 영역을 사용하여 통신비용을 줄임	· 안전 영역의 재설정 비용이 크며, 대부분의 안전 영역의 사이즈가 작음
MQM [15]	×	○	×	· 이동 데이터의 연산 자원 능력을 활용하기 위한 상주 영역(resident domain) 개념 도입 · 각각의 이동 데이터에게 상주 영역을 할당하기 위해 메인 메모리 기반 BP-tree 활용	· 이동 데이터는 새로운 상주 영역의 요청 및 질의 결과 갱신 요청을 위한 메시지만을 송신하므로 서버-이동 데이터 사이의 통신비용을 줄임 · 분산화된 질의 연산을 통해 서버의 질의 처리 비용을 줄임	· 질의의 클리핑(clipping)을 통해 BP-tree를 구축하므로 서버에 등록된 질의의 개수보다 더 많은 개수의 질의 범위를 관리해야 하며, 이동 데이터의 연산 자원 능력을 충분히 활용하지 못함
MobiEyes [16]	○	○	○	· 이동 데이터 및 이동 질의를 메인 메모리 기반 그리드 색인 구조로 색인 · 이동 데이터의 연산 자원 능력을 활용하기 위한 모니터링 영역(monitored region) 개념 도입	· 이동 데이터는 자신의 속도 및 방향이 크게 변화하였을 경우 및 질의 결과 갱신 요청을 위한 메시지만을 송신하므로 서버-이동 데이터 사이의 통신비용을 줄임 · 서버의 질의 처리 비용을 줄임	· 이동 데이터는 자신의 속도 벡터가 크게 변화한 경우에만 자신의 변경된 정보를 서버에게 보고하므로 수시로 이동 데이터의 속도와 방향이 변화하는 동적 환경에서 정확한 결과를 도출해 낼 수 없음
P2P-MQM [21]	○	○	○	· 중앙 서버가 존재하지 않는 모바일 애드혹 혹은 모바일 P2P 환경에서 그리드 색인 구조를 이용하여 연속 질의 처리 지원	· 서버 구축이 힘든 환경에서도 연속 질의 처리를 지원 · 다양한 위치기반 질의 유형 및 연속 질의 유형을 지원	· 서버의 부재로 인해, 이동 데이터의 연산 능력이 시스템 성능을 크게 좌우함



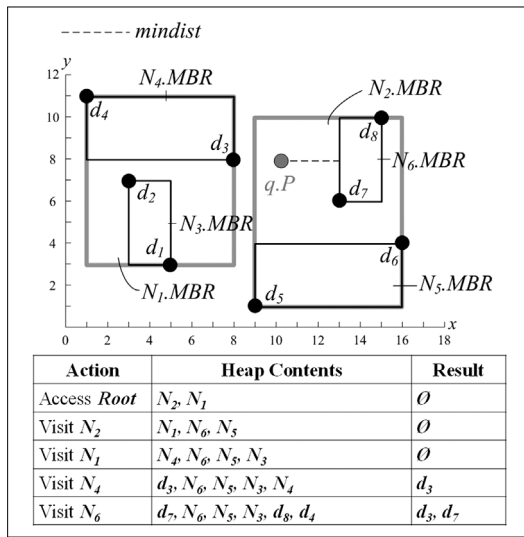


그림 10 R-tree에서 최적 우선 (best-first) 탐색

를 보여준다. 먼저, 알고리즘은 루트 노드에 접근 후 루트 노드가 포함하는  $N_1$ 과  $N_2$ 에 대한 엔트리들을 이들의 mindist와 함께 힙에 삽입한다. 그 후, 엔트리  $N_2$ 에 대한 엔트리를 힙에서 꺼내고  $N_2$ 를 접근하고  $N_2$ 가 포함하는 모든 엔트리들( $N_5$ 와  $N_6$ 에 대한 엔트리들)을 힙에 삽입한다. 이와 같은 과정을 반복하여 질의의 결과인 데이터  $d_3$ 과  $d_7$ 를 검색한 후 종료한다.

### 3.2 연속 k-최근접 질의 처리 기법

연속 k-최근접 질의 처리에 대한 주요 연구들은 이동 데이터 및 질의의 위치가 변할 때마다 단순히 서버에게 자신의 위치를 보고하는 중앙 서버 중심의 질의 처리 연산 시 서버의 CPU-효율적인 질의 처리 연산에 초점을 맞추어 진행되어 왔으며[18-20,22], 최근 서버-이동 데이터 사이의 통신 오버헤드 절감을 위한 연구[23]가 소개되었다. 본 절에서는 연속 k-최근접 질의 처리에 대한 주요 연구들을 소개한다.

#### 3.2.1 Yu-CNN

Yu-CNN.[18]은 이동 데이터에 대한 연속 고정 k-최근접 질의 처리를 위하여 메인 메모리 기반 그리드 색인 구조를 이용한 데이터 색인과 질의 색인을 제안하였다. 데이터 색인 기법에서는 오버홀(overhaul) 알고리즘과 점진적(incremental) 알고리즘을 사용한다. 한편, 질의 색인 기법은 데이터가 아닌 질의를 색인하며 질의의 수가 아주 적을 때만 좋은 성능을 보인다. 따라서 데이터 색인 기법에서 오버홀 알고리즘과 점진적 알고리즘만 소개한다. 이들은 모두 k-최근접 질의의 결과를 주기적으로 재 검토하는 전략을 취한다.

우선 질의의 초기 결과를 검색하기 위해서 질의 점

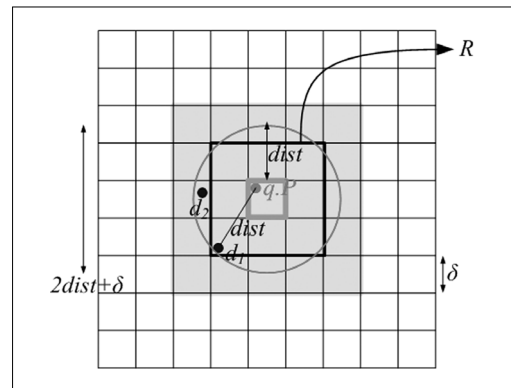


그림 11 1-최근접 질의의 초기 결과 검색의 예

$q.P$ 를 포함하는 셀을 중심으로 사각형  $R$ 을  $k$  개의 데이터를 포함할 때까지 확장시킨다. 이 때, 그림 11과 같이  $R$ 이 질의  $q$ 의 결과를 포함한다고 보장할 수 없다(그림 9에서  $d_1$ 과  $q.P$  사이의 거리를  $dist$ 라고 했을 때  $d_2$ 와  $q.P$  사이의 거리는  $dist$ 보다 짧다). 따라서  $q.P$ 를 포함하는 셀을 중심으로 하고 한 변의 길이가  $2 \cdot dist + \delta$ 인 사각형을 검색 범위로 하여 이와 겹치는 셀들의 집합(그림 11에서 음영 처리 된 부분)을 모두 검색한 후  $d_2$ 를  $q$ 의 결과에 포함시킨다.

데이터 색인 기법에서의 오버홀 알고리즘은 매 주기마다 모든 데이터의 위치를 기반으로 색인 구조를 새로 구축하고 질의를 재 수행한다. 한편, 점진적 알고리즘은 이전 결과에 포함되어 있던 데이터들 중에서 질의 점  $q.P$ 와 현재 가장 거리가 먼 데이터와의 거리인  $dist_{max}$ 를 활용한다. 즉,  $q.P$ 를 포함하는 셀을 중심으로 하고 한 변의 길이가  $2 \cdot dist_{max} + \delta$ 인 사각형을 검색 범위로 하여 이와 겹치는 셀들의 집합을 모두 검색한다. 그림 12는 데이터  $d_2$ 가  $d_{2,old}$ 에서  $d_{2,new}$ 로 이동하였을 경우의 예를 보인다.

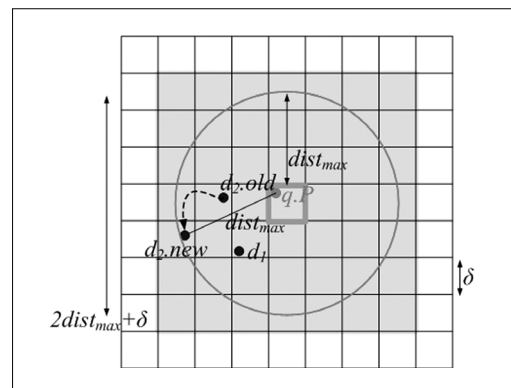


그림 12 1-최근접 질의에 대한 점진적 알고리즘의 예

### 3.2.2 SEA-CNN

SEA-CNN [19]은 디스크 기반 그리드 색인 구조를 사용하여 데이터를 색인한다. 또한 SEA-CNN은 k-최근접 질의의 초기 결과에 대한 연산보다는 모니터링 영역 개념을 활용한 점진적인 결과 갱신에 초점을 맞추며, 연속 이동 k-최근접 질의 처리를 지원한다. 여기서 모니터링 영역이란 질의 점  $q.P$ 를 중심으로 하고  $q.P$ 와 질의  $q$ 의 k번째 결과와의 거리를 반지름으로 하는 원이다. 매 재 수행 주기마다 SEA-CNN은 각각의 질의에 대한 모니터링 영역을 활용하여 다음과 같은 세 단계 연산을 순차적으로 수행 한다:

- 만약 각 질의  $q$ 의 이전 결과에 포함되어 있던 데이터들 중 모니터링 영역 내에서만 이동하였거나 모니터링 영역 외부 데이터가 모니터링 영역 내부로 들어온 경우,  $q$ 의 모니터링 영역을  $q$ 의 재 수행 검색 범위로 설정한다.
  - 만약 각 질의  $q$ 의 이전 결과에 포함되어 있던 데이터들 중 모니터링 영역을 벗어난 경우,  $q$ 의 재 수행 검색 범위는 질의 점  $q.P$ 를 중심으로 하고 반지름이  $dist_{max}$ 인 원으로 설정한다. 여기서  $dist_{max}$ 는 질의 점  $q.P$ 와 현재 가장 거리가 먼 데이터와의 거리이다.
  - 마지막으로, 만약 각 질의 점  $q.P$ 의 위치가 변했을 경우,  $q$ 의 재 수행 검색 범위를  $q.P_{new}$ 를 중심으로 하고 2 단계에서의  $dist_{max}$ 와  $q.P_{new}$ 에서  $q.P_{old}$ 까지의 거리의 합을 반지름으로 하는 원으로 설정한다. 여기서  $q.P_{new}$ 는 질의  $q$ 의 새로운 위치이며  $q.P_{old}$ 는 질의  $q$ 의 이전 위치이다.
3. 위의 세 단계 연산을 모두 수행 후, SEA-CNN은 검색 범위와 겹치는 셀들의 집합을 모두 검색 후 새로운 결과를 검색하고 새로운 모니터링 영역을 설정한다. 그림 13은 데이터  $d_2$ 가  $d_{2,old}$ 에서  $d_{2,new}$ 로 이동하였을 경우의 예를 보인다. SEA-CNN은 그림 13에서 음영 처리 된 셀들을 모두 검색한 후  $d_1$ 을 새로운 결과에 포함시키고 새로운 모니터링 영역을 설정한다.

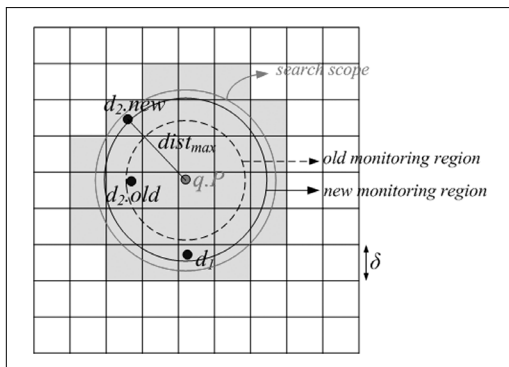


그림 13 1-최근접 질의에 대한 SEA-CNN의 예

### 3.2.3 CPM

CPM(Conceptual Partitioning Monitoring)[20]은 메모리 기반 그리드 색인 구조를 이용하며, 각각의 k-최근접 질의  $q$ 의 위치  $q.P$ 를 포함하는 셀  $c_q$ 를 중심으로 그리드 셀들을 4가지 방향( $Up$ ,  $Down$ ,  $Left$ ,  $Right$ )의 사각형들로 구성하여 전체 도메인  $D$ 를 분할한다. 각각의 사각형  $rect$ 는 방향과 레벨로 정의 된다. 여기서  $rect$ 의 방향은  $q.P$ 와의 위치 관계에 의해  $U$ ,  $D$ ,  $L$ ,  $R$  중 하나로 결정되고,  $rect$ 의 레벨은  $c_q$ 와  $rect$  사이에 존재하는 사각형의 개수이다.

CPM은 질의의 초기 결과 검색을 위하여 먼저 키 값에 대하여 오름차순 정렬 힙을 생성 및 초기화 한다. 힙에 질의 점  $q.P$ 를 포함하고 있는 셀  $c_q$ 와 사각형  $U_0$ ,  $D_0$ ,  $L_0$ ,  $R_0$ 을 키 값과 함께 삽입한다. 여기서 셀  $c_q$ 의 키 값은 0이며  $U_0$ ,  $D_0$ ,  $L_0$ ,  $R_0$ 의 키 값은  $q.P$ 와 최소 거리인  $mindist$  값이다. 그 후 반복적으로 힙에서 첫 번째 엔트리를 삭제하며 다음과 같은 연산을 수행한다:

- 만약 힙에서 삭제한 엔트리가 셀일 경우 셀 안의 데이터들 중 현재 k-최근접 질의 결과에 포함된 데이터보다 더 가까운 데이터가 존재한다면  $q$ 의 결과를 갱신한다.
- 만약 힙에서 삭제한 엔트리가 사각형일 경우 그 사각형을 구성하는 셀들과 다음 레벨(현재 사각형의 레벨 + 1)의 사각형들을  $mindist$  값과 함께 힙에 삽입한다. 위와 같은 연산을 힙의 다음 엔트리 키 값이 현재 k-최근접 질의 결과에 포함된 데이터들 중  $q.P$ 에서부터

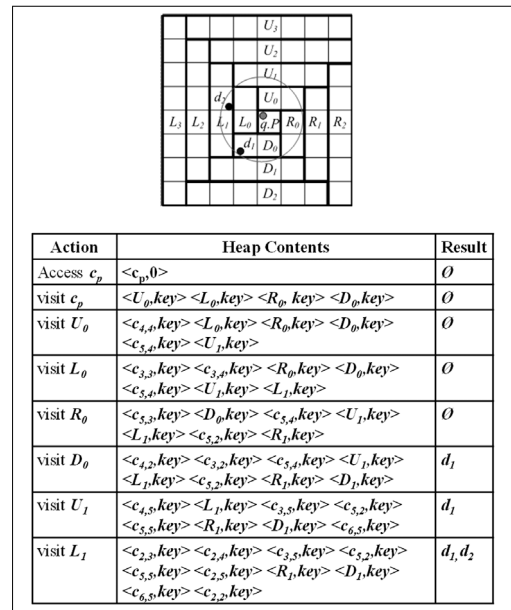


그림 14 2-최근접 질의에 대한 CPM의 예

가장 먼 데이터의 거리  $kth\_dist$ 보다 크거나 같으면 종료된다. 그 후, 점진적인 결과 갱신을 위해  $q.P$ 를 중심으로 하고 반지름이  $kth\_dist$ 인 원을 모니터링 영역으로 설정한다. 그림 14는 CPM의 k-최근접 질의 초기 결과 검색 및 모니터링 영역 설정을 보인다.

그 후, CPM은 모니터링 영역 내로 이동한 데이터들의 수를 체크하고 모니터링 영역 안에서 밖으로 이동한 데이터들의 수가 더 많다면 질의 결과를 다시 갱신 하는데, 얼마나 많은 수의 이동 데이터가 빠져나가는지 모니터링 한 후 빠져나간 데이터의 개수가 삽입된 데이터의 개수보다 많으면 최근접 질의 결과를 다시 구하게 되고 그렇지 않으면 모니터링 영역안의 데이터들만으로 질의 결과를 갱신하고 모니터링 영역을 재설정 한다.

### 3.2.4 iSEE

iSEE[22]는 메인 메모리 기반 그리드 색인구조를 이용하며, 질의 점  $q.P$ 에서부터 각 그리드 셀까지의 최소 거리인  $mindist$  값이 유사한 셀들끼리 레벨을 설정하고 그룹화 하여 연속 k-최근접 질의를 처리하는 VOB (visit order builder)를 제안하였다. 그림 15는 VOB를 위한 레벨 설정 및 그룹화 방법을 보여준다. 질의 점  $q.P$ 를 포함하고 있는 셀  $c_0$ 를 레벨 0으로 설정하여 질의 점에서 가장 인접한(즉,  $mindist$  값이 가장 작은) 주변 셀들을 레벨 1로 설정, 그 다음 인접한 주변 셀들을 레벨 2로 설정하는 방식으로 각각의 셀의 레벨을 설정한 뒤, 각 레벨 별로  $c_0$ 의 중심점으로부터의  $mindist$ 가 같은 셀들을 하나의 그룹으로 하여 그룹화 한다.

iSEE는 질의의 초기 검색 결과를 얻기 위하여  $c_0$ 와 그룹  $L_1G_1$ (1 레벨, 1 그룹)을 키 값  $mindist$ 와 함께 오름차순 정렬 힙에 삽입한 후(여기서  $L_iG_j$ 의  $mindist$  값은  $L_iG_j$ 에 속한 각 셀들의  $mindist$ 중 가장 작은 값으로 설정한다.), iSEE는 반복적으로 힙에서 첫 번째 엔트리를 삭제하며 CPM과 마찬가지로 다음과 같은 연산을 수행한다:

만약 힙에서 삭제한 엔트리가 셀일 경우 해당 셀 안의 데이터들 중 현재 k-최근접 질의 결과에 포함된 데이터보다 더 가까운 데이터가 존재한다면  $q$ 의 결과를 갱신한다.

만약 힙에서 삭제한 엔트리가 그룹화된 셀들의 집합  $L_iG_j$ 일 경우  $L_iG_j$ 에 속한 셀들을 각 셀들과  $L_{i+1}G_j$ 를  $mindist$ 와 함께 힙에 삽입한다. 한편,  $i = j$ 일 경우에는 추가적으로  $L_iG_{j+1}$ 를  $mindist$ 와 함께 힙에 삽입한다.

위와 같은 연산을 힙의 다음 엔트리 키 값이 현재 질의 결과 데이터들 중  $q.P$ 에서부터 가장 먼 데이터의 거리인  $kth\_dist$ 보다 크거나 같으면 종료하며,  $kth\_dist$ 를 반지름으로 하는 원을 갱신 연산을 위한 모니터링 영역으로 설정한다. 그림 15는 iSEE의 k-최근접 질의의 초

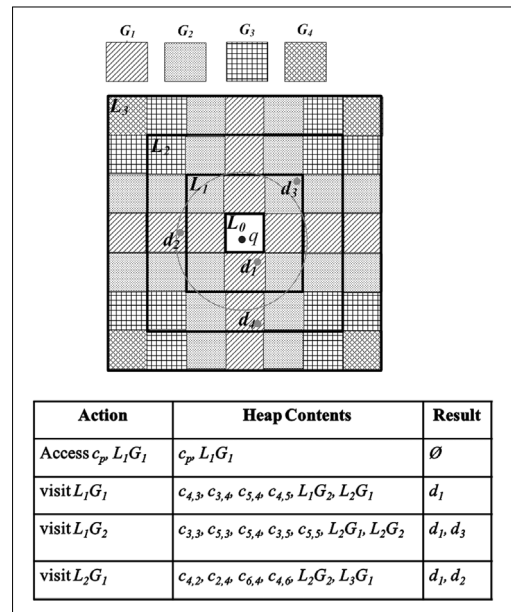


그림 15 2-최근접 질의에 대한 iSEE의 예

기 결과 검색 및 모니터링 영역 설정을 보인다.

그 후, iSEE는 CPM과 마찬가지로 모니터링 영역 내부로 들어온 데이터의 개수가 외부로 나간 데이터의 개수보다 많을 경우와 모니터링 영역 내부로 들어온 데이터의 개수가 외부로 나간 데이터의 개수보다 적을 경우로 나누어 갱신연산을 수행하고 모니터링 영역을 재설정 한다.

### 3.2.5 disMKNN

disMKNN[23]은 중앙 서버 중심의 질의 처리 연산이 아닌 이동 데이터의 연산 자원을 바탕으로 분산화 된 환경에서 이동 데이터에 대한 연속 이동 k-최근접 질의 처리 기법을 제안한다. 특히, disMKNN은 하나의 기지국(base station)이 수용할 수 있는 범위를 하나의 셀로 정의한다. disMKNN의 k-최근접 질의 처리는 초기 결과 검색 연산과 갱신 연산으로 나누어진다. 질의가 요청 되었을 경우 초기 결과를 검색하고, 해당 질의가 종료되기 전까지 지속적으로 갱신 연산 처리과정을 수행한다:

1. 초기 결과 검색 연산: 새로운 k-최근접 질의 생성자는 자신의 위치  $q.P$ 와 속도 벡터를 서버에 전송한다. 서버는  $q.P$ 가 속한 셀 및 인접한 셀들을 조사하여 k개의 임시 결과 값을 찾고, 결과 값에 포함된 데이터들 중  $q.P$ 에서부터 가장 먼 데이터의 거리인 임시  $kth\_dist$ 를 반지름으로 하는 원과 겹치는 셀들에게 질의 생성자의 정보를 전송한다. 정보를 받은 해당 셀들의 데이터들은 질의 점  $q.P$ 와 자신의 거리  $dist$  값을 계산하여  $kth\_dist$  값보다 작으면 서버에 자신의

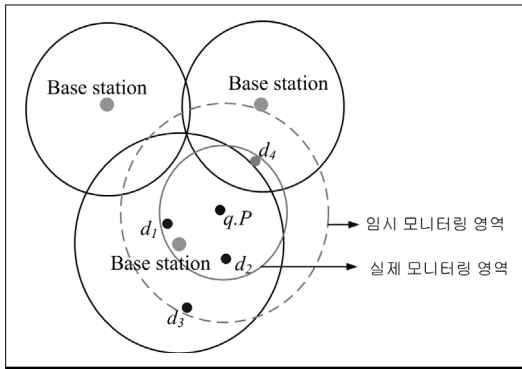


그림 16 disMKNN의 초기 결과 검색의 예

위치 및 속도를 전송한다. 서버는 전송 받은 정보를 바탕으로 실제 k-최근접 질의 결과 값을 반환한다. 또한, 질의 결과 값 중 k번째 데이터  $d$ 와  $q.P$  사이의 거리를 실제  $kth\_dist$ 로 설정하고, 질의 생성자와  $d$ 의 위치 및 속도 벡터를  $q.P$ 를 중심으로 하고  $kth\_dist$ 를 반지름으로 하는 원과 겹치는 셀(Influenced Cells:  $IC(q.P)$ )들에 포함된 데이터들에게 전송한다. (서버와 데이터들 간의 정보 전송은 기지국을 통해서 이루어지므로 서버가 모든 데이터들을 관리할 필요가 없다.) 이러한 초기 결과 검색이 끝난 후 disMKNN은 지속적 갱신 연산을 질의가 종료되는 시점까지 수행한다. 그림 16은 disMKNN의 초기 결과 검색의 예를 보여준다.

2. 지속적 갱신 연산:  $IC(q.P)$  내의 데이터들은 자신의 연산 능력을 바탕으로 질의 점  $q.P$ 와 k번째 데이터  $d$ 의 속도 벡터를 이용하여 자신이 질의의 결과에 포함되는 시간을 계산하여 해당 시간에 서버에 자신의 위치 및 속도 벡터를 전송한다. 그 후 서버는 질의 결과를 갱신하고 새로운 k번째 질의 결과 데이터  $d'$ 을  $IC(q.P)$ 에 전송한다. 해당 데이터들은 새로운 결과를 바탕으로 질의 결과 포함 시간을 구한다(만약 질의 생성자 및 k번째 질의 결과 데이터의 속도 벡터가 변화하게 되면 서버는  $IC(q.P)$  내의 데이터들에 변화된 속도를 전송하여  $IC(q.P)$  내의 데이터들은 변화된 속도를 바탕으로 자신이 질의 결과에 포함될 시간을 구한다.). 이러한 과정 중에 서버는 데이터 및 질의의 이동으로 인한  $IC(q.P)$ 의 변화를 (1)  $IC(q.P)$ 에 겹치는 혹은 포함되는 셀들과  $IC(q.P)$ 의 반경, (2) 질의 생성자와 k번째 데이터의 위치 및 속도 벡터를 바탕으로  $IC(q.P)$ 에 속하는 셀의 변경 시간을 구하여, 해당 시간에  $C(q.P)$ 를 갱신하여 해당 셀에 질의 생성자의 위치  $q.P$ 와 k번째 질의 결과 데이터의 위치 및 속도 벡터를 지속적으로 재전송한다.

### 3.2.5 P2P-MQM

2.2.5절에서 언급하였듯이, P2P-MQM[21]은 이동 데이터에 대한 연속 고정 k-최근접 질의 및 이동 데이터에 대한 연속 이동 k-최근접 질의의 처리가 가능하다. P2P-MQM은 이동 데이터에 대한 연속 고정 k-최근접 질의를 처리하기 위해 질의 생성자의 위치와 k번째 가까운 데이터의 거리와  $k+1$ 번째 가까운 데이터의 거리의 평균을 반지름으로 하는 원을 모니터링 영역으로 설정하고 이 영역을 범위 질의로 간주하여 2.2.5절에서 소개한 질의 처리 기법을 통해 수행한다(그림 17(a) 참조). 또한, 이동 데이터에 대한 연속 이동 k-최근접 질의 처리 역시 2.2.5절에서 설명한 이동 데이터에 대한 연속 이동 범위 질의 처리와 같이  $Bf$ ,  $Bi$ ,  $Bo$ 를 설정하여 처리한다(그림 17(b) 참조).

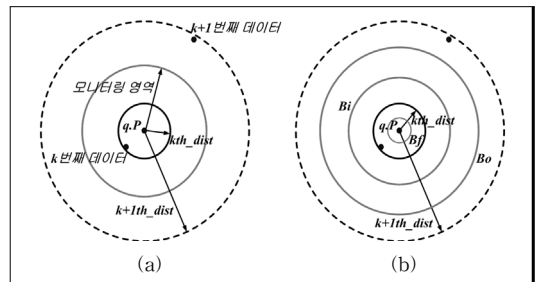


그림 17 P2P-MQM의 k-최근접 질의 처리의 예

### 3.3 비교분석

3장에서는 연속 k-최근접 질의 처리를 위한 연구 Yu-CNN[18], SEA-CNN[19], CPM[20], iSEE[22], disMKNN[23], P2P-MQM[21]을 소개하였다. Yu-CNN, SEA-CNN, CPM, iSEE는 이동 데이터가 자신의 위치가 변할 때마다 단순히 서버에게 자신의 새로운 위치를 보고하므로 서버와 이동 데이터 사이의 통신비용이 크고, 서버 중심의 질의 처리 연산으로 인해 이동 데이터 혹은 질의의 수가 증가하게 되면 서버의 병목현상이 발생할 수 있다. 한편, disMKNN은 이동 데이터의 연산 자원을 바탕으로 서버와 협력적으로 질의 처리를 수행하므로 서버의 복잡한 자료구조 및 알고리즘이 불필요하며 질의 및 이동 데이터들은 자신의 이동으로 인하여 질의 결과 값에 영향을 미칠 수 있을 경우에만 서버에 자신의 새로운 정보를 보고하므로 통신비용을 줄일 수 있다. P2P-MQM은 연속 범위 질의 처리와 마찬가지로 중앙서버가 존재하지 않는 환경에서의 그리드 색인 구조를 이용한 질의 처리를 지원한다. 표 2는 위에서 소개한 연속 k-최근접 질의 처리를 위한 연구 내용에 대한 요약 및 비교·분석을 보이고 있다. 또한, 각 기법의 성능 평가는 해당 기법의 논문들을 참조하길 바란다.

표 2 연속 k-최근접 질의 처리 주요 연구

주요 연구	연속 질의 유형			주요 내용	장점	단점
	고정 데이터 / 이동 질의	이동 데이터 / 고정 질의	이동 데이터 / 이동 질의			
Yu-CNN [18]	×	○	×	<ul style="list-style-type: none"> <li>메인 메모리 기반 그리드 색인 구조를 활용한 데이터 색인 및 질의 색인 기법</li> <li>데이터 색인 기법 중 오버홀 질의 재수행 알고리즘과 점진적 질의 재수행 알고리즘 제안</li> </ul>	<ul style="list-style-type: none"> <li>색인 구조의 갱신연산 및 질의 연산 비용을 줄임</li> </ul>	<ul style="list-style-type: none"> <li>이동 데이터는 자신의 위치가 변할 때마다 단순히 서버에게 보고하므로 서버-이동 데이터 사이의 통신비용이 큼</li> <li>이동 데이터 혹은 질의 수가 증가하게 되면 서버의 병목현상 발생</li> </ul>
SEA-CNN [19]	△	○	△	<ul style="list-style-type: none"> <li>디스크 기반 그리드 색인 구조를 활용하며 질의와 데이터를 색인</li> <li>점진적 질의 재수행을 위한 모니터링 영역 개념 도입</li> </ul>	<ul style="list-style-type: none"> <li>모니터링 영역을 활용한 점진적 질의의 재수행을 통해 질의 연산 비용을 줄임</li> </ul>	<ul style="list-style-type: none"> <li>서버-이동 데이터 사이의 통신비용이 큼</li> <li>서버의 병목현상 발생</li> <li>메인 메모리 기반 색인구조보다 검색시간이 오래 걸림</li> <li>이동 데이터에 대한 연속 이동 질의 처리 비용이 상당히 큼</li> </ul>
CPM [20]	×	○	×	<ul style="list-style-type: none"> <li>메인 메모리 기반 그리드 색인 구조를 활용</li> <li>각 질의의 위치를 중심으로 전체 그리드 도메인을 가상의 사각형으로 분할</li> <li><i>mindist</i>를 활용한 best-first 알고리즘 제안</li> </ul>	<ul style="list-style-type: none"> <li>다양한 자료구조와 <i>mindist</i>를 활용한 질의 재 수행을 통해 질의 연산 비용을 줄임</li> </ul>	<ul style="list-style-type: none"> <li>서버-이동 데이터 사이의 통신비용이 큼</li> <li>서버의 병목현상 발생</li> <li>다양한 자료구조로 인한 메모리 사용이 큼</li> </ul>
iSEE [22]	×	○	×	<ul style="list-style-type: none"> <li>메인 메모리 기반 그리드 색인 구조를 활용</li> <li>각 질의의 위치를 중심으로 <i>mindist</i> 값을 활용한 그리드 셀의 레벨 설정 및 그룹화를 통한 VOB 기법 제안</li> </ul>	<ul style="list-style-type: none"> <li>CPM과 마찬가지로 다양한 자료구조와 <i>mindist</i>를 활용한 질의 재 수행을 통해 질의 연산 비용을 줄임</li> </ul>	<ul style="list-style-type: none"> <li>서버-이동 데이터 사이의 통신비용이 큼</li> <li>서버의 병목현상 발생</li> <li>다양한 자료구조로 인한 메모리 사용이 큼</li> </ul>
disMKNN [23]	○	○	○	<ul style="list-style-type: none"> <li>분산화된 환경에서 이동 데이터의 연산 자원 능력을 활용</li> <li>질의 결과 값에 영향을 미칠 가능성이 있는 셀들을 리스트화한 Influenced Cells 사용</li> </ul>	<ul style="list-style-type: none"> <li>서버에 복잡한 자료구조 및 알고리즘이 불필요</li> <li>결과 값에 영향을 미칠 수 있는 경우에만 서버와 이동 데이터가 통신하므로 통신비용을 줄임</li> <li>갱신 연산시 질의 결과 값에 영향을 미칠 가능성이 있는 셀들만 갱신 연산 수행에 관여함으로써 연산 비용을 줄임</li> </ul>	<ul style="list-style-type: none"> <li>기지국의 반경 범위에 의해 셀의 범위가 설정되므로 셀의 크기가 시스템 성능을 크게 좌우함</li> <li>셀의 범위가 원으로 설정되므로 모든 영역을 수용하기 위하여 각 셀들 끼리 겹치는 영역이 발생</li> </ul>
P2P-MQM [21]	○	○	○	<ul style="list-style-type: none"> <li>중앙서버가 존재하지 않는 모바일 애드혹 혹은 모바일 P2P 환경에서 그리드 색인 구조를 이용하여 연속 질의 처리 지원</li> </ul>	<ul style="list-style-type: none"> <li>서버 구축이 힘든 환경에서도 연속 질의 처리를 지원할 수 있음</li> <li>다양한 위치기반 질의 유형 및 연속 질의 유형을 지원</li> </ul>	<ul style="list-style-type: none"> <li>서버의 부재로 인해, 이동 데이터의 연산 능력이 시스템 성능을 크게 좌우함</li> </ul>

#### 4. 향후 연구 방향

지금까지 가장 대표적인 위치기반 질의인 범위 질의와 k-최근접 질의 처리에 대한 주요 연구 현황을 알아보았다. 이를 기반으로 본 장에서는 위치기반 질의의 향후 연구 방향을 도출하고자 한다. 첫 번째로 범위 질의

와 k-최근접 질의 혹은 다양한 위치기반 질의를 통합적으로 처리할 수 있는 연구가 필수적이다. 대부분의 위치기반 질의 처리 기법들의 효율성은 특정 질의 유형에 국한되어 있다. 앞서 설명한 P2P-MQM의 경우와 같이 범위 질의와 k-최근접 질의를 통합적으로 처리할 수 있는 연구는 점차 증대될 것으로 예측한다. 두 번째로, 최

근 소개되고 있는 새로운 유형의 위치기반 질의[24-26]에 대한 연구는 일회성 질의의 효율적인 처리에 초점을 맞추고 있다. 하지만 최근 스마트폰의 등장과 다양한 무선통신 기술의 발전으로 인해 질의 및 데이터의 이동성을 고려한 질의 처리 기술에 대한 요구는 증가하고 있으며, 이러한 요구를 충족시키기 위하여 새로운 유형의 위치기반 질의 의 연속 질의 처리 기법에 관한 연구가 필요하다.

마지막으로, 기존의 위치기반 질의는 사용자 (질의)와 데이터 사이의 공간적인 관계만을 고려하므로, [예제 1]과 같이 다양한 비공간 속성을 고려하여 사용자가 선호하는 정보를 제공하는 실질적인 위치기반 정보 서비스를 지원하기에는 한계를 지닌다.

**예제 1.** 그림 18과 같이 위치와 메뉴의 평균 가격을 포함하는 지역 레스토랑 데이터의 집합  $R = \{r_1, r_2, \dots, r_8\}$ 을 가정하자. 더불어, 위치기반 정보 서비스에 가입한 사용자가 가까운 레스토랑을 검색한다고 가정하자. 기존의 위치기반 서비스는 위치 정보만을 활용하므로 사용자의 현재 위치( $q.P$ )에서 가장 가까운 레스토랑  $r_7$ 의 정보를 제공할 것이다. 하지만, 사용자가 선호하는 메뉴의 평균 가격이 \$10라고 할 때 레스토랑  $r_7$ 은 사용자에게 무의미한 정보일 수 있다.

또한, 연속 위치 기반 질의는 질의 및 데이터의 이동성으로 인해 지속적이고 빠른 질의 결과 유효성 검증이 필수적이다. 이는 서버의 오버헤드 및 서버-이동 데이터 혹은 서버-이동 사용자 사이의 통신 오버헤드를 증가시킨다. 이러한 환경에서 비공간 속성을 고려한 위치기반 질의는 불필요한 정보에 대한 연산을 줄임으로써 서버 및 통신 오버헤드를 경감시킬 수 있다. 그림 19는 데이터의 비공간 속성을 고려한 MQM의 효율성을 보여주는 예이다. 이처럼 데이터의 비공간 속성을 고려함으로써 이동 데이터의 상주 영역이 커지게 되고 불필요한 질의를 처리할 필요가 없게 된다. 위와 같이 공간 속성뿐만

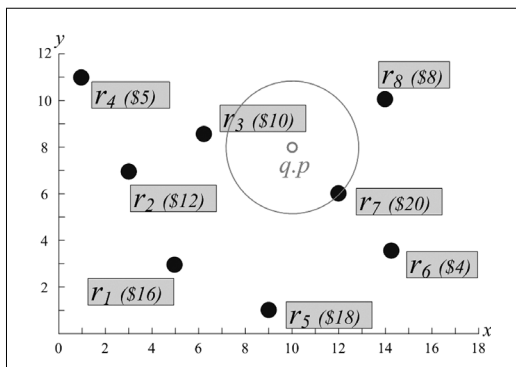


그림 18 샘플 레스토랑 데이터

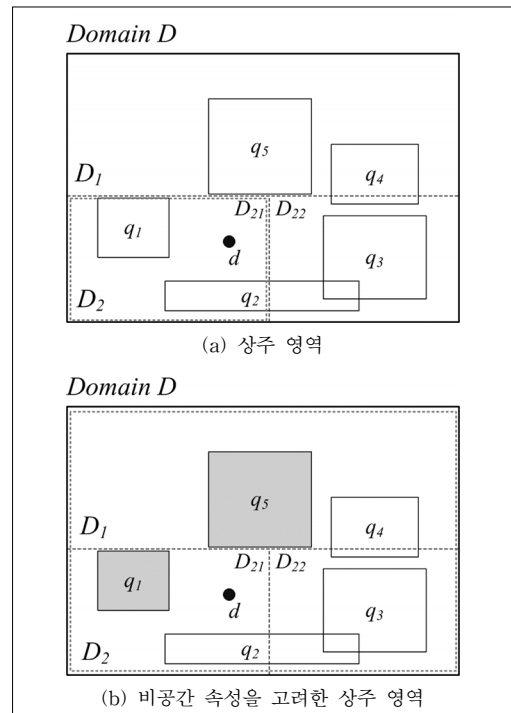


그림 19 MQM에서 비공간 속성을 고려한 상주 영역

아니라 비공간 속성도 고려하는 고급화된 위치기반 질의에 대한 요구로 인해 최근 관련 연구[24,27-29]가 등장하고 있는 추세이고 점차 증대될 것으로 예측한다.

## 5. 결 론

최근 모바일 기기 및 통신 환경의 발전으로 위치기반 서비스에 대한 사용자의 요구는 점차 증가되고 있다. 본 논문은 효과적인 위치기반 정보 서비스 제공의 초석이 될 수 있는 위치기반 질의 처리 기법의 동향을 소개·분석하였다. 특히, 질의 및 데이터의 위치 변화에 따라 지속적으로 질의 결과의 유효성을 검증하고 새로운 결과를 신속하고 에너지 효율적으로 제공하기 위한 연속 위치기반 질의 처리 기법에 초점을 맞추었다. 위치기반 질의 처리에 대하여 범위 질의와 k-최근접 질의 처리 연구에 대하여 크게 두 가지로 구분하였다. 세부적으로 고정 데이터에 대한 일회성 질의와 연속 질의로 나누어 해당 질의 처리 연구들의 동향을 소개·분석하였다. 또한, 범위 질의와 k-최근접 질의의 통합적 질의 처리와 새로운 질의들에 대한 연속 질의의 모니터링 기법의 필요성을 역설하였고, 데이터의 공간적인 속성만을 고려하는 기존의 위치기반 질의의 한계점을 제시하고, 그에 따른 해결책으로 비공간 속성을 고려한 위치기반 질의 처리 연구의 방향을 제시하였다.

## 참 고 문 헌

- [1] A. Guttman, "R-trees: a dynamic index structure for spatial searching," *In SIGMOD Conference*, pp.47-57, 1984.
- [2] Roussopoulos N., Kelley S., Vincent F., "Nearest Neighbor Queries," *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pp.71-79, May. 1995.
- [3] R. Benetis, C. S. Jensen, G. Karciuskas, and S. Saltenis, "Nearest Neighbor and Reverse Nearest Neighbor Queries for Moving Objects," *In Proc. IDEAS*, pp.44-53, 2002.
- [4] D. Papadias, Q. Shen, Y. Tao, and K. Mouratidis, "Group nearest neighbor queries," *In ICDE*, pp.301-312, 2004.
- [5] H. Hu and D. L. Lee, "Range Nearest-Neighbor Query," *IEEE TKDE*, pp.78-91, 2006.
- [6] K.L. Cheung and A. Wai-Chee Fu, "Enhanced Nearest Neighbor Search on the R-Tree," *SIGMOD Record*, vol.27, no.3, pp.16-21, 1998.
- [7] Hjaltason, G., Samet, H., "Distance Browsing in Spatial Databases," *TODS*, pp.265-318, 1999.
- [8] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-Tree: A Dynamic Index for Multi-Dimensional Objects," *Proc. 13rd Int'l Conf. Very Large Data Bases*, pp.507-518, Sept. 1987.
- [9] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R\*-Tree: An Efficient and Robust Access Method for Points and Rectangles," *In Proc. ACM SIGMOD*, pp.322-331, 1990.
- [10] Tao, Y., Papadias, D., "Spatial Queries in Dynamic Environments," *ACM TODS*, pp.101-139, 2003.
- [11] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez, "Indexing the positions of continuously moving objects," *In SIGMOD Conference*, 2000.
- [12] Y. Tao, D. Papadias, and J. Sun, "The TPR\*-Tree: An Optimized Spatio-Temporal Access Method for Predictive Queries," *In Proc. VLDB*, pp.790-801, 2003.
- [13] S. Prabhakar, Y. Xia, D. Kalashnikov, W. Aref, and S. Hambrusch, "Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects," *IEEE Transactions on Computers, Special section on data management and mobile computing*, 51(10), 2002.
- [14] D. V. Kalashnikov, S. Prabhakar, and S. E. Hambrusch, "Main memory evaluation of monitoring queries over moving objects," *Distrib. Parallel Databases*, 15(2):117-135, 2004.
- [15] Cai Y., K.A. Hua, G. Cao and Xu T., "Real-Time Processing of Range-Monitoring Queries in Heterogeneous Mobile Databases," *IEEE Transactions on Mobile Computing*, pp.931-942, 2005.
- [16] Gedik B. and Liu L., "Mobieyes: Distributed Processing of Continuously Moving Queries on Moving Objects in a Mobile System," *Proceedings EDBT Conference*, pp.67-87, 2004.
- [17] Mokbel M., Xiong X. and Aref W., "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases," *Proceedings of ACM SIGMOD Conference*, pp.623-634, 2004.
- [18] Yu X., Pu K.Q. and Koudas N., "Monitoring K-Nearest Neighbor Queries over Moving Objects," *Proceedings of ICDE Conference*, pp.631-642, 2005.
- [19] Xiong X., Mokbel M., Aref W., "SEA-CNN: Scalable Processing of Continuous K-Nearest Neighbor Queries in Spatio-temporal Databases," *Proceedings of ICDE Conference*, pp.643-654, 2005.
- [20] K. Mouratidis, M. Hadjieleftheriou, and D. Papadias, "Conceptual partitioning: An efficient method for continuous nearest neighbor monitoring," *Proceedings of ACM SIGMOD Conference*, pp.634-645, 2005.
- [21] Galdames, P., Kim, K., and Cai, Y., "A Generic Platform for Efficient Processing of Spatial Monitoring Queries in Mobile Peer-to-Peer Networks," *In MDM*, 2010.
- [22] Wu, W., Tan, K. L., "iSEE: Efficient continuous k-nearest-neighbor monitoring over moving objects," *In SSDBM*, p.36, 2007.
- [23] W. Wu, W. Guo, and K.-L. Tan, "Distributed processing of moving k-nearest-neighbor query on moving objects," *In IEEE International Conference on Data Engineering (ICDE)*, pp.1116-1125, 2007.
- [24] Hua Lu, Man Lung Yiu, "On Computing Farthest Dominated Locations," *IEEE Transactions on Knowledge and Data Engineering*, 2010.
- [25] Yunjun Gao, Baihua Zheng, Gencai Chen, Qing Li, "Optimal-Location-Selection Query Processing in Spatial Databases," *IEEE Transactions on Knowledge and Data Engineering*, vol.21, no.8, pp.1162-1177, 2009.
- [26] K. C. K. Lee, W. C. Lee, and H. V. Leong, "Nearest surrounder queries," *In Proceedings of the ICDE Conf.*, pp.85-94, 2006.
- [27] H. Jung, B. Cho, Y. D. Chung and L. Liu, "On processing location based top-k queries in the wireless broadcasting system," *ACM SAC*, pp. 585-591, 2010.
- [28] I. De Felipe, V. Hristidis, and N. Rishe., "Keyword search on spatial databases," *In ICDE*, pp.656-665, 2008.
- [29] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. *In ICDE*, pp.688-699, 2009.



이 성 민

2010년 고려대학교 컴퓨터통신공학부 학사  
2010년~현재 고려대학교 컴퓨터전과통신공학과 석사과정. 관심분야는 Spatio-temporal Databases, Location-based Services 등



정 하 림

2004년 광운대학교 컴퓨터학과 학사. 2007년 고려대학교 컴퓨터학과 석사. 2007년~현재 고려대학교 컴퓨터학과 박사과정  
관심분야는 Mobile/Pervasive Databases, Spatio-temporal Databases, Location-based Information Systems 등



정 연 돈

1994년 고려대학교 전산학과 졸업(학사). 1996년 한국과학기술원 전산학과 졸업(석사). 2000년 한국과학기술원 전산학 전공 졸업(박사). 2000년~2003년 한국과학기술원 전산학전공 Post-Doc. 연구원 및 연구교수. 2003년~2006년 동국대학교 컴퓨터공학과 교수. 2009년~2009년 Georgia Tech. 방문교수. 2006년~현재 고려대학교 컴퓨터·통신공학부 교수. 관심분야는 Database Privacy, Spatial Databases, Mobile Databases, Graph Databases, Data-Intensive Systems, Database Systems



이 기 용

1998년 2월 KAIST 전산학과 학사. 2000년 2월 KAIST 전산학과 석사. 2006년 2월 KAIST 전자전산학과 전산학전공 박사. 2006년 3월~2008년 2월 삼성전자 기술총괄 소프트웨어연구소 책임연구원  
2008년 3월~2009년 8월 KAIST 전산학 전공 연구조교수. 2009년 9월~현재 숙명여자대학교 컴퓨터 과학과 교수. 관심분야는 Data warehouse, OLAP, Embedded DB