

# 이동 객체 환경에서 거리 관계 패턴 기반 $k$ -최근접 질의 처리 기법

(A  $k$ -NN Query Processing Method based on Distance Relation Patterns in Moving Object Environments)

박용훈<sup>†</sup> 서동민<sup>\*\*</sup> 북경수<sup>\*\*\*</sup> 이병엽<sup>\*\*\*\*</sup> 유재수<sup>\*\*\*\*\*</sup>  
 (Yong Hun Park) (Dong Min Seo) (Kyoung Soo Bok) (Byoung Yup Lee) (Jae Soo Yoo)

**요약** 최근 유클리드 공간 상에서 효율적인  $k$ -최근접( $k$ -Nearest Neighbors) 질의 처리를 위해 그리드 구조 기반의 많은 색인 기법들이 연구되었다. 하지만 기존 기법들은  $k$ -최근접 객체들을 연산하기 위해 불필요한 셀을 접근하여 연산 자원을 낭비하거나 근접한 셀을 알아내는데 매우 큰 연산 비용을 초래한다. 그래서 본 논문에서는 한 셀과 주변 셀과의 거리 관계를 나타내는 거리 관계 패턴을 이용하여  $k$ -최근접 질의 처리시 적은 연산 비용과 적은 저장 공간을 사용하는 새로운  $k$ -최근접 질의 처리 기법을 제안한다. 본 논문에서는  $k$ -최근접 질의 처리의 대표적인 기법인 CPM과 성능을 비교하여 제안하는 기법의 우수성을 입증한다.

**키워드** : 이동 객체, 위치 기반 서비스,  $k$ -최근접 질의 처리

**Abstract** Recently, various methods have been proposed to process  $k$ -NN ( $k$ -Nearest Neighbors) queries efficiently. However the previous methods have problems that they access additional cells unnecessarily and spend the high computation cost to find the nearest cells. In this paper, to overcome the problems, we propose a new method to process  $k$ -NN queries using the patterns of the distance relationship between the cells in a grid. The patterns are composed of the relative coordinates of cells sorted by the distance from certain points. Since the proposed method finds the nearest cells to process  $k$ -NN queries with traversing the patterns sequentially, it saves the computation cost. It is shown through the various experiments that our proposed method is much better than the existing method, CPM, in terms of the query processing time and the storage overhead.

**Key words** : Moving Object, LBS(Location Based Service),  $k$ -NN Query Processing

\* 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업 그리고 2009년 교육과학기술(지역거점연구단육성사업/충북BIT연구중심 대학육성사업단)부로부터 지원받아 수행된 연구임

<sup>†</sup> 학생회원 : 충북대학교 정보통신공학과  
yhpark@netdb.chungbuk.ac.kr

<sup>\*\*</sup> 학생회원 : KAIST 전산학과  
dmseo@chungbuk.ac.kr

<sup>\*\*\*</sup> 정회원 : 가인정보기술  
ksbok@netdb.chungbuk.ac.kr

<sup>\*\*\*\*</sup> 정회원 : 배재대학교 전자상거래학과 교수  
bylee@pcu.ac.kr

<sup>\*\*\*\*\*</sup> 종신회원 : 충북대학교 전기전자컴퓨터공학부 교수  
yjs@chungbuk.ac.kr  
(Corresponding author)

논문접수 : 2008년 11월 28일

심사완료 : 2009년 4월 10일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이때, 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 제36권 제3호(2009.6)

## 1. 서론

휴대용 모바일 기기의 발전과 위치 인식 기술의 발전으로 인하여 위치 기반 서비스에 대한 관심이 급격하게 증가하고 있다. 이동 객체를 대상으로 하는 질의 처리 기법은 향상된 위치 기반 서비스를 제공하기 위해 기술적인 관점에서 매우 중요한 부분으로서 시스템의 성능을 좌우하는 요소중의 하나이다.

$k$ -최근접( $k$ -Nearest Neighbor) 질의는 임의의 질의 점으로부터  $k$ 개의 가장 근접한 객체들을 찾는 질의이다.  $k$ -최근접 질의는 많은 위치 기반 서비스에서 응용된다. 예를 들면, 콜택시 회사에서 택시를 원하는 고객으로부터 가장 근접한 택시를 알아내서 고객에게 제공할 수 있다. 또한 객체가 이동하는 환경에서는 시간이 지남에 따라  $k$ -최근접 질의의 결과가 지속적으로 변하기 때문에 특정 시간 동안 결과를 모니터링 하는 연속  $k$ -최근

접 질의 위치 기반 서비스에서 필수적이다.

YPK-CNN[1]은 영역 질의 기법을  $k$ -최근접 질의 처리에 적용한 기법이다.  $k$ -최근접 객체들을 찾기 위해 질의 점으로 포함하는 셀을 중심으로 각 방향으로 검색할 셀들을 한 겹씩 확장한다. YPK-CNN은 사각형을 기반으로 하고, 각 방향으로 동등하게 확장하므로 불필요한 셀을 많이 방문하여 처리 비용이 낭비된다. 불필요한 셀 탐색을 방지하기 위해 제안된 CPM[2]은 가지치기(branch and bound) 알고리즘의 best-first 기법을 트리 구조가 아닌 그리드 구조에 적용한 기법이다. 하지만 CPM은 연속 질의 처리를 위해 방문 리스트와 가지치기 알고리즘을 위해 사용했던 힙을 질의마다 유지하여 저장 공간의 낭비를 초래한다.

이러한 단점들을 극복하고 연속  $k$ -최근접 질의 처리를 효율적으로 처리하기 위해서 본 논문에서는 거리 관계 패턴을 새롭게 제안한다. 거리 관계 패턴은 한 셀 내의 임의의 점으로부터 주변의 다른 셀들을 거리에 대한 오름차순으로 정렬하고 그 셀들을 그 점이 존재하는 셀을 기준으로 상대 좌표로 나타낸 셀 좌표 리스트들의 셋이다. 이 패턴은 동일한 크기의 셀로 구성된 그리드에서 유효하다. 그리드 색인에서  $k$ -최근접 질의를 처리하기 위해 가장 많이 차지하는 연산 비용 중의 하나인 근접한 셀을 찾는 연산을 최소화하기 위해 거리 관계 패턴이 사용된다. 또한 거리 관계 패턴을 고려한 질의 색인구조와  $k$ -최근접 질의 처리 기법을 제안한다. 그리고 본 논문에서는  $k$ -최근접 질의 처리 시 거리 값을 기준으로 정렬된 거리 관계 패턴의 상대좌표를 순차적으로 적용하여 근접한 셀을 알아내기 때문에 우선 순위 큐를 이용하는데 드는 연산비용을 절감하였다. 또한 거리 관계 패턴 외 추가적인 자료구조를 유지하지 않기 때문에 저장 공간의 낭비를 방지한다. 마지막으로 제안하는 질의 처리 기법의 성능을 검증하기 위해 연속  $k$ -최근접 질의 처리의 대표적인 기법인 CPM과 비교를 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 기술한다. 3장에서는 거리 관계 패턴을 정의하고, 제안하는 질의 처리 기법과 질의 색인 구조를 기술한다. 4장에서는 제안하는 질의 처리 기법과 CPM에 대한 분석적인 비교를 보여준다. 5장에서는 성능 평가한 결과를 보여주고, 6장에서는 결론을 다룬다.

## 2. 관련 연구

이동 객체 환경에서의 연속 질의 처리 기법은 크게 연속 영역(range) 질의와 연속  $k$ -최근접 질의로 분류된다. 연속 영역 질의 처리를 위한 디스크 기반 색인 기법으로는 R-트리와 안정 영역(safe region) 개념을 이용한 Q-Index[3], 질의의 이동을 함께 고려한 SINA[4] 그리고

질의 영역을 구분하기 위해 color 개념을 도입한 cGridex[5]가 제안되었다. 그리고 메모리 기반 색인 기법으로는 셀 기반 색인인 CQI[6] 그리고 가상 셀 기반 색인인 VCRI[7]와 CES[8]가 제안되었다. 연속  $k$ -최근접 질의 처리를 위한 디스크 기반의 기법으로  $k$ 개의 객체를 보장하는 영역을 계산하고 그 영역과 겹치는 셀들을 모두 탐색하는 SEA-CNN[8]이 제안되었고, 메모리 기반의 색인 기법으로는 질의 점을 둘러싸는 셀들을 한 겹씩 검사하는 YPK-CNN[1], 질의 점을 중심으로 각 방향의 셀들을 그룹화하고 우선순위 큐를 이용한 CPM[2]이 제안되었다.

본 논문은 연속  $k$ -최근접 질의 처리를 위한 메인 메모리 기반의 색인 기법을 제안하기 때문에 YPK-CNN과 CPM에 대해서 자세히 설명한다. YPK-CNN은 영역 질의 처리 기법을  $k$ -최근접 질의 처리에 적용한 기법으로  $k$ -최근접 객체들을 찾기 위해 셀 영역을 확장하면서 질의 처리를 수행한다. 주어진 질의 점을 포함하는 셀을 탐색하고, 만약 그 셀에  $k$ 개의 객체가 존재하지 않으면, 그 셀을 둘러싸는 주변의 셀들을 탐색한다. 검색 영역 확장을 통한 탐색은  $k$ -최근접 객체를 보장하는  $k$ 개의 객체를 찾을 때까지 수행된다. 그림 1(a)에서 YPK-CNN이 2-최근접 질의 처리시 탐색하는 셀을 어두운 색으로 나타냈다. 연속 질의 처리시 객체가 질의 점으로부터  $k$ 번째 객체까지의 거리  $k$ -dist를 기준으로 밖에서 안으로 들어온 경우 질의의 결과 집합의  $k$ -최근접 객체와 비교하여 결과 집합과  $k$ -dist를 갱신하여 점진적인 처리를 수행한다. 하지만 객체가  $k$ -dist 안에서 밖으로 벗어나거나 또는 질의 점이 이동을 한 경우 질의를 점진적으로 처리 하지 않고 질의를 재 수행한다. YPK-CNN은 사각형을 기반으로 셀을 탐색 하고, 각 방향으로 동등하게 확장하므로 불필요한 셀을 많이 방문하여 처리 비용이 낭비된다. 또한 연속 질의 처리시  $k$ -dist를 기준으로 객체가 밖에서 안으로 들어올 경우에만 점진적인 처리를 수행하여 연속 질의 처리의 효율성이 낮다.

CPM은 YPK-CNN의 불필요한 셀 탐색 문제와 연속 질의의 비효율성의 문제를 해결하기 위해 제안되었다. CPM 기법은 가지치기(branch and bound) 알고리즘의 best-first 기법을 트리 구조가 아닌 그리드 구조에 적용한 기법이다. 이 기법은 우선 순위 힙을 이용하여 항상 질의 점으로부터 가장 근접한 셀을 순서대로 찾는 것을 보장한다. 질의에 사용되는 메모리 사용량을 줄이기 위해 그리드 영역의 모든 셀을 힙에 삽입하지 않고, 질의 점을 중심으로 한 주변 셀들을 가상으로 그룹화하여 삽입한다. 그룹은 트리 구조의 색인관점에서 중간 노드와 같다. 초기 힙에는 질의 점을 포함하는 셀과 그 주변 그룹들이 삽입된다. 그림 1(b)에서 CPM이 2-최근접 질의 처리시 탐색하는 셀을 어두운 색으로 나타냈다.

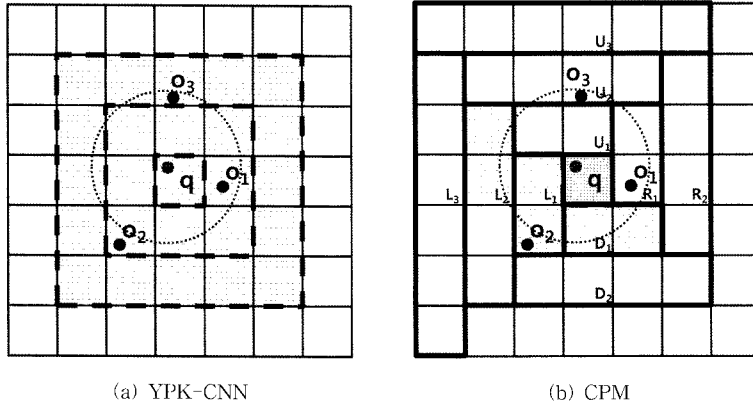


그림 1 YPK-CNN와 CPM을 이용한 2-최근접 질의 처리시 탐색하는 셀

연속 질의 처리를 위해 셀 방문 리스트와 질의 처리를 위해 사용했던 힙을 유지한다. 객체가  $k$ -dist 안으로 이동한 경우 YPK-CNN과 같은 방식으로 점진적인 질의 처리를 수행한다. 객체가  $k$ -dist 밖으로 이동한 경우, 셀 방문 리스트를 이용하여 근접한 셀을 순서대로 탐색한 후에 추가적인 셀 탐색이 필요할 경우 힙을 이용하여 점진적인 질의 처리를 수행한다. 셀 방문 리스트를 이용하기 때문에 근접한 셀을 알아내기 위해 소모되는 연산비용을 감소시켰다. 만약 질의 점이 이동하면 셀 방문 리스트와 힙이 무효화 되므로 질의가 재 수행 된다. 그림 1(b)에서 어두운 색으로 표시된 셀들은 모두 셀 방문 리스트에 유지되는 셀들이고, 그룹  $D_2$ ,  $R_2$ ,  $L_3$  그리고  $U_3$ 는 내부의 흰색으로 표시된 셀들이 정렬되어 힙에 유지되는 그룹이다.

그림 1에서 볼 수 있듯, CPM이 YPK-CNN의 불필요한 셀 탐색 문제를 해결하였다. 하지만 CPM은 연속 질의 처리를 위해 셀 방문 리스트와 힙을 질의마다 유지하여 저장 공간의 낭비를 초래한다. 이로 인한 저장 비용은 질의 점으로부터 탐색 범위와 질의 수에 비례하여 증가한다. 또한, 질의 처리시 힙을 유지하기 위해 소모되는 비용도 무시할 수 없는 부분이다.

### 3. 제안하는 $k$ -최근접 질의 처리 기법

본 논문에서 제안하는 색인 기법은 동일한 크기를 가지는 셀간의 거리 관계 패턴을 이용하여 질의 처리시 적은 비용으로 근접한 셀들을 알아낸다. 그리고 한 점과 셀과의 거리 측정은 CPM과 동일한 방식을 이용한다. 표 1은 본 논문에서 사용하는 기호들을 나타낸다.

#### 3.1 거리 관계 패턴

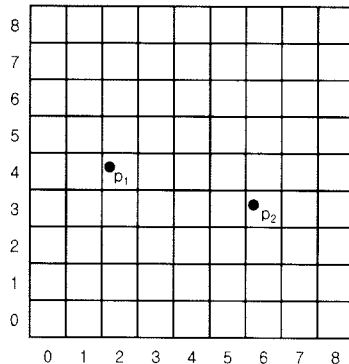
거리 관계 패턴은 한 셀 내의 임의의 점으로부터 다른 셀들을 거리에 대한 오름차순으로 정렬하고 그 셀들을 상대 좌표로 나타낸 셀 좌표 리스트이다. 거리 관계 패턴은 동일한 크기의 셀들이 동일한 간격으로 구성된 형태의 색인 구조에서  $k$ -최근접 질의 처리를 위해 사용될 수 있다.

**정의 1.** 그리드 상의 위치가 다른 두 점  $p_n, p_m$ 에 대해서 만약  $PC(p_n) = PC(p_m)$ 이면,  $CLA(p_n) \neq CLA(p_m)$  이지만  $CLR(p_n) = CLR(p_m)$ 이다.

정의 1은 다른 두 점이 셀 내에서의 상대 좌표가 동일하다면 주변 셀과의 상대적인 거리 관계도 동일하다는 것을 보장한다. 그림 2는 정의 1의 예를 보여준다. 그림 2(a)는 다른 셀에 존재하지만 셀 안에서의 위치는 동일

표 1 기호 설명

기호	설명
$PC(p)$	점 $p$ 가 포함되는 셀 내에서의 상대 좌표
$CLA(p)$	점 $p$ 로부터 셀들을 근접한 순으로 절대좌표로 정렬하여 표현한 리스트
$CLR(p)$	점 $p$ 로부터 셀들을 근접한 순으로 상대좌표로 정렬하여 표현한 리스트
$N_{dir}$	한 셀 내에서 $CLR$ 의 기준이 되는 점들의 개수
$N_{sort}$	한 $CLR$ 에서 유지하는 상대 좌표의 개수
$k$ -dist	질의 결과 셋에서 $k$ 번째 객체와 질의 점과의 거리
$CP(v)$	영역 $v$ 의 중심점
$MINDIST()$	인자로 입력된 두 점 또는 점과 셀과의 최소 거리
$MAXDIST()$	인자로 입력된 점과 셀간의 최대 거리



(a) 임의의 두 점 위치 표현

Closest Cell list of $p_1$		Closest Cell list of $p_2$	
$CLA(p_1)$	$CLR(p_1)$	$CLA(p_2)$	$CLR(p_2)$
C(2,4)	(0,0)	C(6,3)	(0,0)
C(1,4)	(-1,0)	C(5,3)	(-1,0)
C(2,5)	(0,+1)	C(6,4)	(0,+1)
C(1,5)	(-1,+1)	C(5,4)	(-1,+1)
C(2,3)	(0,-1)	C(6,2)	(0,-1)
...	...	...	...

(b) 각각 두 점에 대한 CLA와 CLR차이

그림 2 CLA와 CLR의 차이

한 임의의 두 점  $p_1$ 과  $p_2$ 를 나타내고, 그림 2(b)는 각 두 점의 CLA와 CLR을 나타낸다. 두 점이 존재하는 셀이 다르기 때문에  $CLA(p_1)$ 과  $CLA(p_2)$ 는 다르다. 하지만  $CLR(p_1)$ 과  $CLR(p_2)$ 는 동일한 것을 확인할 수 있다.

CLR은 CLA와는 다르게 셀 내에서의 상대좌표만 동일 하다면 모든 셀에서 동일하게 만들어진다. 이러한 특성을 이용하여 셀 내의 모든 점에 대해서 CLR을 유지한다면 별도의 연산 없이 근접한 셀들을 차례로 접근할 수 있다. 하지만 셀 내의 모든 점에 대해서 CLR을 유지한다는 것은 거의 불가능 하다. 공간에서 점의 개수는 매우 많기 때문이다. 그래서 셀 내에서 대표적으로 의미 있는 점들을 추출하여 그 점들을 기준으로 CLR을 유지한다. 이 CLR들의 집합을 본 논문에서는 거리 관계 패턴으로 정의한다. 정의 2에서 그 거리 관계 패턴을 설명한다.

**정의 2.**  $N_{ctr}$ 은 거리 관계 패턴이 가지는 CLR의 개수이다. 셀 영역을 가상으로  $N_{ctr}$  만큼 분할하고, 그 분할된 작은 셀을  $VDC(Virtually Divided Cell)$ 라고 한다. 한  $VDC v$ 의 중심점은  $CP(v)$ 로 표현하고,  $CLR(CP(v))$ 는 그  $v$  영역을 대표하는 CLR이다.

한  $CLR(CP(v))$ 가 나타내는 CLR  $clr$ 은 배열구조를 가지며  $clr[n]$ 은 점  $CP(v)$ 로부터  $n$ 번째 근접한 셀의

상대 좌표를 나타낸다. 정의 3에서 그 관계를 설명한다.

**정의 3.** 주어진  $VDC v$ 에 대해서 CLR은 식 (1)과 같은 속성을 갖는다.

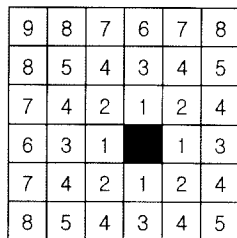
$$MINDIST(cp, clr[i]) \leq MINDIST(cp, clr[i+1])$$

$$\text{단, } cp = CP(v), clr = CLR(cp) \tag{1}$$

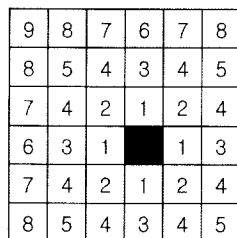
그림 3은  $N_{ctr}$ 의 값에 따라 셀 내의 분할된 영역과 만들어진 CLR의 정확성을 보여준다. 그림 3(a)는  $N_{ctr}$ 이 1이기 때문에 셀의 중심점이 그 CLR의 기준점이 되고 그 영역을 대표한다. 그림 3(b)와 3(c)는  $N_{ctr}$ 이 각각 4와 36이기 때문에 영역을 4개와 36개로 분할한다. 그리고 분할된 영역의 중심점이 각 CLR의 기준점이 된다. 분할된 영역별로 CLR을 만들고 그 영역에 포함되는 점을 기준으로 근접한 셀들을 찾을 때는 그 CLR을 따른다.

거리 관계 패턴의 최대 오차는  $N_{ctr}$ 에 의해 분할된 한 영역  $VDC v$ 내의 모든 점들과 중심점인  $CP(v)$ 로부터 다른 셀들까지의 거리 측정 시 발생할 수 있는 최대 거리 차이이다. 그 오차를 정의 4에서 설명한다.

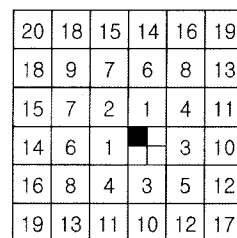
**정의 4.** 그리드 내의 셀들의 집합을  $C=(c_1, c_2, \dots, c_n)$ 로 나타내고, 주어진 한 영역  $VDC v$  내의 점들의 집합을  $P_v=(p_1, p_2, \dots, p_n)$ 로 나타낸다. 최대 오차  $E$ 는 식 (2)와 같다.



(a)  $N_{ctr} = 1$



(b)  $N_{ctr} = 4$



(c)  $N_{ctr} = 36$

그림 3  $N_{ctr}$ 에 따른 거리 관계 패턴

$$E(C, v, P_i) = \max\{|MINDIST(CP(v), c_i) - MINDIST(p_i, c_i)|\}$$

단,  $c_i \in C, p_i \in P_i$  (2)

**정의 5.** 2차원 평면에서 세 개의 점  $p_1, p_2$  그리고  $p_3$  가 있을 때, 삼각 부등식에 의해  $|MINDIST(p_1, p_3) - MINDIST(p_2, p_3)| \leq MINDIST(p_1, p_2)$ 은 보장된다. 만약 임의의 사각형  $B$ 가 있을 때,  $|MINDIST(p_1, B) - MINDIST(p_2, B)| \leq MINDIST(p_1, p_2)$ 을 만족한다.

정의 5는 한 VDC  $v$ 에 존재하는 두 점  $p_i$ 와  $p_j$ 가 있을 때,  $E \leq \max\{MINDIST(p_i, p_j)\}$ 를 보장한다. 그래서 거리 관계 패턴의 최대 오차는  $v$ 에 포함된 점들 중  $CP(v)$ 로부터 가장 멀리 떨어진 점까지의 거리이다.  $a$ 가 한 셀의 측면 길이라고 할 때, 최대 오차  $E$ 는 식 (3)과 같고, 그림 4는  $N_{clr}$ 이 4일 때, 거리 관계 패턴의 최대 오차  $E$ 를 보여준다. 어떻게 표시된 VDC  $v$ 에서  $CP(v)$ 와 모서리의 한 점  $p$ 까지 거리가 최대 오차  $E$ 를 나타낸다.

$$E = \alpha\sqrt{2} / (2 \times \sqrt{n_{clr}}) \quad (3)$$

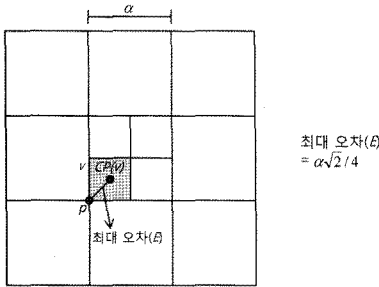


그림 4 최대 오차 E 계산

이러한 오차 때문에 거리 관계 패턴은 한 VDC내의 모든 점들에 대해서 만족되지 않고 일부의 점들에 대해서만 만족된다. 이러한 관계를 정의 6에서 설명한다.

**정의 6.** 주어진 VDC  $v$  내의 점들의 집합을  $P_v = \{p_1, p_2, \dots, p_n\}$ 로 나타낸다.  $cp = CP(v)$ ,  $clr = CLR(cp)$ 라고 가정한다. 조건  $MINDIST(p_i, clr[k]) + MINDIST(p_i, cp) < MINDIST(cp, clr[k])$ 을 만족하는 점  $p_i$ 에 대해서는 조건  $MINDIST(p_i, clr[k]) < MINDIST(cp, clr[k + \beta])$ 이 성립되어 거리 관계 패턴이 유효하지만, 그렇지 않은 점  $p_i$ 에 대해서는 거리 관계 패턴이 유효하지 않다. 단,  $\beta \geq 0$ 를 만족하는 정수이다.

거리 관계 패턴을 기반으로 영역  $v$ 에 존재하는 임의의 점  $p_i$ 로부터 근접한 셀들을 순차적으로 접근할 때,  $p_i$ 와  $CP(v)$ 의 거리 차이로 인해 정확성을 보장하지 못할 수도 있다. 거리 오차는  $MINDIST(p_i, CP(v))$ 보다 작으므로 원하는 검색 반경을  $D_{want}$ 라고 할 때, 실제 검색 반경  $D_{search}$ 는 식 (4)와 같다.

$$D_{search} = D_{want} + MINDIST(p_i, CP(v)) \quad (4)$$

거리 관계 패턴은 두 개의 인자를 이용하여 생성된다. 하나는 정렬할 셀의 개수  $N_{sort}$ 이고, 나머지 하나는  $CLR$ 의 기준이 되는 점들의 개수  $N_{clr}$ 이다.  $N_{sort}$ 와  $N_{clr}$ 은 거리 관계 패턴이 차지하는 저장 공간에 영향을 미친다. 또한  $N_{clr}$ 은 거리 관계 패턴의 정확성에 영향을 미친다. 거리 관계 패턴의 저장 비용은 정의 7에서 설명한다.

**정의 7.** 거리 관계 패턴의 저장 비용은  $N_{sort}$ 와  $N_{clr}$ 에 비례한다. 한 좌표를 표현하기 위한 저장 비용을  $S_{crd}$ 로 나타낼 때, 거리 관계 패턴을 위한 저장 비용  $S_{pat}$ 는 식 (5)와 같다.

$$S_{pat} = S_{crd} \times N_{sort} \times N_{clr} \quad (5)$$

거리 관계 패턴을 이용하면 적은 연산 비용으로 근접한 셀들을 순서대로 접근할 수 있다. 그리고 거리 관계 패턴의 오차를 줄이기 위해  $N_{clr}$ 을 크게 지정할 수 있지만  $N_{clr}$ 의 증가는 저장 공간의 소모를 초래하기 때문에 적절한 값을 결정해야 한다.

### 3.2 k-최근접 질의 처리 기법

본 논문에서 제안하는 패턴 기반의 k-최근접 질의 처리 기법은 3.1절에서 정의한 거리 관계 패턴을 이용한다. 거리 관계 패턴은  $CLR$ 의 집합으로써 셀 내에서의 질의 점의 좌표에 따라 질의에 사용될  $CLR$ 이 결정된다. 그  $CLR$ 을 이용하여 k-최근접 질의 처리를 수행한다. 표 2는 질의 처리 알고리즘을 표현하기 위한 기호들을 나타낸다.

질의 처리는 3가지 단계로 수행된다. 첫 번째로  $N_{sort}$ 와  $N_{clr}$ 을 인자로 하여 거리 관계 패턴을 생성한다. 거리 관계 패턴은 그리드를 구성하는 셀의 모양과 배치를 기반으로 해서 만들어지기 때문에 한번 생성하면 영구적으로 사용 가능하다. 또한 거리 관계 패턴은  $CLR$ 들을 관리하고 좌표를 해시 키로 사용한다. 두 번째로 질의를 생성하고  $CLR$ 을 할당 받는다. k-최근접 질의는 질의 점, 검색할 객체의 개수, 결과 객체, 할당 받은  $CLR$  정보를 갖는다. 거리 관계 패턴으로부터 질의의 점을 키로 적절한  $CLR$ 을 할당 받는다. 세 번째로 할당 받은  $CLR$ 을 이용하여 k-최근접 질의를 처리한다.  $CLR$ 의 상대 좌표를 순서대로 참고 하여 k개의 객체가 검색될 때까지 근접한 셀을 순서대로 탐색한다. 우선 질의 점이 존재하는 셀을 탐색하고, 발견된 객체가 k개 미만이면,  $CLR$ 의 첫 번째 상대 좌표로 다음 셀을 알아내고 탐색한다. 발견된 객체가 k개 미만이면, 다시  $CLR$ 의 두 번째 상대 좌표로 다음 셀을 알아내고 탐색한다. 이러한 방식으로 k객체를 찾을 때까지 셀을 탐색한다. 이때 질의 점에서 k번째 객체까지의 거리를 k-dist라고 한다. 검색된 k개의 객체가 k-최근접 객체임을 보장하는지 확인하기 위해 추가적으로 k-dist보다 질의 점에 가까운 모든 셀들의 탐색을 순서대로 진행한다. 만약 추가적으로 발견된 객체가 k-dist보다 질의 점에 더 가까우면,

표 2 알고리즘 표현을 위한 기호

기호	설명
$a$	한 셀의 측면 길이
$k$	최근접 객체의 수
$HCLR(q)$	질의 $q$ 의 좌표를 포함하는 VDC를 $v$ 라고 할 때, $CLR(CP(v))$
$QC(q)$	질의 $q$ 를 포함하는 셀
$AC(c, rc)$	셀 $c$ 를 기준으로 상대 좌표 $rc$ 에 존재하는 셀
$OL(c)$	셀 $c$ 의 객체 리스트
$QL(c)$	셀 $c$ 의 질의 리스트
$q.x, q.y$	공간 상에서 질의 $q$ 의 $x$ 축의 좌표와 $y$ 축의 좌표
$q.result\_set$	질의 $q$ 의 결과 집합
$q.pclr$	질의 $q$ 가 탐색한 CLR상의 위치를 나타내는 값
$q.k-dist$	질의 $q$ 와 $q.result\_set$ 내의 $k$ 번째 근접한 객체간 거리
$q.qid$	질의 $q$ 의 식별자
$clr[i]$	임의 CLR $clr$ 의 $i$ 번째 상대좌표
$clr[i].x$	임의 CLR $clr$ 의 $i$ 번째 $x$ 축의 상대좌표
$clr[i].y$	임의 CLR $clr$ 의 $i$ 번째 $y$ 축의 상대좌표
$clr[i].dist$	질의 $q$ 의 좌표를 포함하는 VDC를 $v$ 라고 할 때, $MINDIST(CP(v), clr[i])$

결과 셋과  $k-dist$ 를 갱신한다. 남은 셀들이  $k-dist$ 보다 질의 점에 가까운 것이 없으면 처리를 중단한다.

그림 5는 3-최근접 질의를 처리하는 과정을 나타낸다. 세번째 단계에서  $k$ 개의 객체를 찾는 과정을 그림 5(a)에서, 그 이후  $k$ -최근접 객체임을 보장하기 위한 과정을 그림 5(b)에서 나타낸다.

하지만 CLR에는 오차가 있기 때문에 추가적인 셀 탐색이 필요하다. CLR상의 다음 순서에 탐색할 셀들이 이전 순서에 탐색한 셀들보다 질의 점에 근접할 수 있다. 이것은 CLR이 담당하는 영역에 존재하는 모든 점에 대해서 근접한 셀의 순서를 만족 시키지 않기 때문이다. 오차를 극복하기 위해  $search-dist$ 를 이용하여 셀을 탐색한다.  $EQ(q)$ 는 질의  $q$ 에 대해서  $k-dist$ 의 오차 값을 나타낼 때,  $search-dist$ 는 식 (6)과 같다.

$$search\_dist(q) = q.k-dist + EQ(q) \quad (6)$$

그림 6은  $E$ 와  $EQ(q)$ 의 차이 그리고  $search-dist$ 와

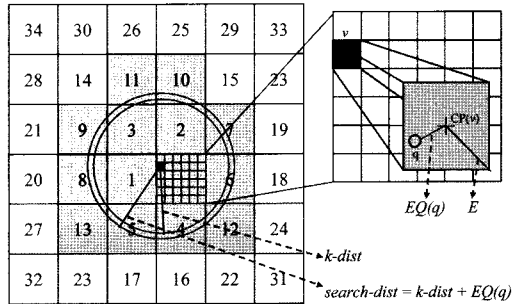
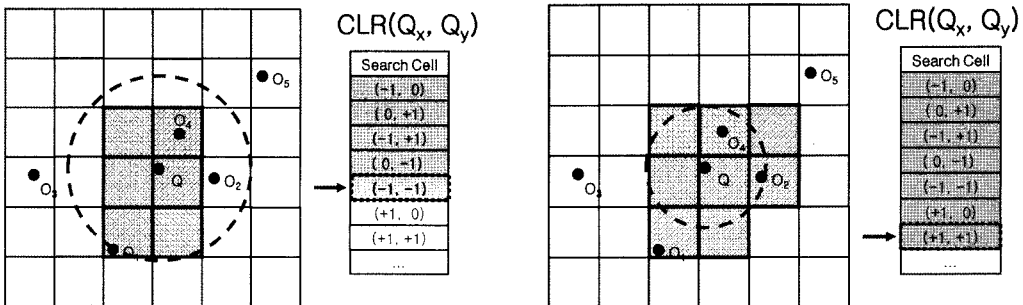


그림 6 search-dist를 고려한 검색 영역

$k-dist$ 의 차이를 보여준다.  $k$ -최근접 객체를 보장하기 위해 CLR에서 제공하는 순서로 셀 탐색 시,  $search-dist$ 와 탐색하는 셀과의 거리를 비교하면 거리 관계 패턴의 오차가 고려된다. 하지만 실제 셀을 탐색할 때는 질의 점으로부터의 거리가  $k-dist$ 보다 작은 셀만을 탐색한다.



(a) CLR을 이용한 3개의 객체 탐색

(b) 3-NN을 보장하는 객체 탐색

그림 5 거리 관계 패턴을 이용한 3-최근접 질의 처리

```

KNN_Computation(a)
// Input = query q
// Output = query result
01 clr = HCLR(a)
02  $C_q = QC(a)$ 
03 While ( $k > |a.result\_set|$ )
04    $C_{cur} = AC(C_q, clr[q.p_{clr}])$ 
05   For each object  $o \in OL(C_{cur})$ 
06     Insert  $o$  into a.result_set:
07    $q.p_{clr}++$ 
08   Update  $q.k-dist$ 
09   While ( $search\_dist(a) > clr[q.p_{clr}].dist$ )
10     If ( $q.k-dist < MINDIST(a, C_{cur})$ )
11        $q.p_{clr}++$ 
12       continue
13      $C_{cur} = AC(C_q, clr[q.p_{clr}])$ 
14     For each  $o, \forall o \in OL(C_{cur})$ 
15       If ( $q.k-dist > MINDIST(o, q)$ )
16         Remove  $o^{old}$  from a.result_set
17         Insert  $o$  into a.result_set
18         Update  $q.k-dist$ 
19      $q.p_{clr}++$ 
20   Return a.result_set
    
```

그림 7  $k$ -최근접 질의 처리 알고리즘

그림 7은  $k$ -최근접 질의를 연산하는 알고리즘이다. 일단 주어진 질의  $q$ 가 내려지면 질의  $q$ 에 해당되는 CLR  $clr$ 을 구하고(줄 01), 질의  $q$ 의 좌표를 포함하는 셀  $C_{cur}$ 을 구한다(줄 04).  $q.p_{clr}$ 을 1씩 증가시켜 가면서  $clr[q.p_{clr}]$ 을 셀  $C_{cur}$ 의 좌표에 적용하여 근접한 셀을 순서대로 탐색하고,  $k$ 개의 객체를 찾아  $result\_set$ 에 삽입한다(줄 05~07).  $k$ 개의 객체를 찾은 후  $q.k-dist$ 를 계산한다(줄 08).  $k$ 개의 객체가  $k$ -최근접 객체임을 보장하기 위해  $q.p_{clr}$ 을 1씩 증가시켜 가면서  $clr[q.p_{clr}]$ 에 따라  $search\_dist(q) > clr[q.p_{clr}].dist$  조건을 만족할 때까지 근접한 셀들을 추가적으로 탐색한다(줄 09). 만약  $q.k-dist$ 보다 근접한 객체가 발견되면  $q.result\_set$  그리고  $q.k-dist$ 를 모두 갱신한다(줄 14~18).

#### 4. 분석

이 장에서는 대표적인  $k$ -최근접 질의 처리 기법인 CPM과 제안하는 질의 처리기법인 PB-KNN의 연산비용을 비교 분석한 결과를 설명한다.

##### 4.1 PB-KNN의 연산 비용

PB-KNN이  $k$ -최근접 질의 처리 시 요구하는 연산 비용  $C_{pb-knn}$ 은 식 (7)에서 나타낸다.

$$C_{pb-knn} = C_{dist\_cells} + C_{dist\_objects} + C_{adapt\_drp} \quad (7)$$

$C_{pb-knn}$ 은 질의와 셀과의 거리를 연산하는 비용  $C_{dist\_cells}$ 와  $k$ -최근접 객체를 확인하기 위해 질의와 객체와의 거리를 연산하는 비용  $C_{dist\_objects}$  그리고 근접한 셀을 찾기 위해 거리 관계 패턴의 상대 좌표를 적용하는 비용  $C_{adapt\_drp}$ 으로 구성된다.

$C_{dist\_cells}$ 는 한 셀과 질의와의 거리 측정 비용을  $C_{dist\_a\_cell}$

라고 하고, 질의 처리에 의해 접근된 셀의 개수를  $N_{visit\_cells}$ 라고 할 때, 식 (8)과 같다.

$$C_{dist\_cells} = C_{dist\_a\_cell} \times N_{visit\_cells} \quad (8)$$

$C_{dist\_objects}$ 는 한 객체와 질의와의 거리 측정 비용을  $C_{dist\_a\_object}$ 로 하고, 한 셀의 포함되는 평균 객체의 개수를  $N_{objects\_in\_cell}$ 로 표기할 때, 식 (9)와 같다.

$$C_{dist\_objects} = C_{dist\_a\_object} \times N_{objects\_in\_cell} \times N_{visit\_cells} \quad (9)$$

$C_{adapt\_drp}$ 은 질의를 포함하는 셀의 좌표에 CLR상의 상대 좌표를 적용하는 비용을  $C_{adapt\_coordinate}$ 라고 할 때, 식 (10)에서 표현한다.

$$C_{adapt\_drp} = C_{adapt\_coordinate} \times N_{visit\_cells} \quad (10)$$

결국  $C_{pb-knn}$ 은 식 (11)과 같이 표현한다.

$$C_{pb-knn} = N_{visit\_cells} (C_{dist\_a\_cells} + C_{dist\_a\_object} \times N_{objects\_in\_cell} + C_{adapt\_coordinate}) \quad (11)$$

$k$ 개의 최근접 객체를 구하기 위해 탐색하는 셀의 개수  $N_{visit\_cells}$ 는 식 (12)와 같다.

$$N_{visit\_cells} = \left\lceil \frac{k}{N_{objects\_in\_cell}} \times \alpha \right\rceil \quad (\text{단, } \alpha \geq 1) \quad (12)$$

$N_{visit\_cells}$ 는  $k$ 개의 객체를 포함하는 면적과 비례하기 때문에 그 면적에 계수  $\alpha$ 를 곱하여 구한다.

객체의 수를  $|O|$ 로 나타내고, 그리드의 셀 개수를  $|C|$ 로 나타낸다.  $N_{objects\_in\_cell}$ 은 식 (13)과 같다.

$$N_{objects\_in\_cell} = \frac{|O|}{|C|} \quad (13)$$

그래서  $N_{visit\_cells}$ 는 식 (14)와 같다.

$$N_{visit\_cells} = \left\lceil \frac{k \cdot |C|}{|O|} \times \alpha \right\rceil \quad (14)$$

식 (12), 식 (13) 그리고 식 (14)를  $C_{pb-knn}$ 에 대한 식 (11)에 대입하면 식 (15)으로 표현된다.

$$C_{pb-knn} = \left[ \frac{k \cdot |C|}{|O|} \times \alpha \right] (C_{dist\_a\_cells} + C_{dist\_a\_object} \times \frac{|O|}{|C|} + C_{adapt\_coordinate}) \quad (15)$$

위의 식은  $k$ 값과 연산 비용은 비례한다는 것을 나타낸다. 하지만  $|O|$ 와  $|C|$ 에 의해서 결정되는  $N_{objects\_in\_cell}$ 이 너무 크거나 작으면 연산비용의 증가를 초래한다. 최적의 성능을 위해서는  $|O|$ 와  $|C|$ 에 대해서 적절한 값의 결정이 필요하다.

##### 4.2 CPM의 연산 비용

CPM이  $k$ -최근접 질의 처리 시 요구하는 연산 비용  $C_{cpm}$ 은 식 (16)에서 나타낸다.

$$C_{cpm} = C_{dist\_cells} + C_{dist\_objects} + C_{priority\_queue} \quad (16)$$

$C_{pb-knn}$ 와 다른 점은  $C_{adapt\_drp}$ 의 부재와  $C_{priority\_queue}$ 의 추가이다.  $C_{priority\_queue}$ 는 CPM이 질의 처리시에 우선순위 큐를 유지하는데 소모되는 비용이다. 식 (17)에서 그 비용을 나타낸다.

$$C_{priority\_queue} = C_{sort} \times N_{visit\_cells} \quad (17)$$

$C_{sort}$ 는 정렬 비용을 나타낸다. 우선 순위 큐는 매번 우선순위를 결정하기 위해서 정렬을 수행한다. CPM에서는  $k$ -최근접 질의를 처리하기 위해서 탐색한 셀의 개수만큼 정렬을 수행한다. 그 정렬 비용을 주된 비용으로 고려한다.

### 4.3 PB-kNN과 CPM의 연산 비용 비교

PB-kNN과 CPM의 차이를 명확히 하기 위해서 PB-kNN이 CPM에 비해서 절약하는 연산 비용  $C_{save}$ 를 식 (18)에서 표현한다.

$$C_{save} = (C_{sort} - C_{adapt\_coordinate}) \times N_{visit\_cells} \quad (18)$$

$C_{sort}$ 는 정렬비용이기 때문에 평균 비용은  $O(n \cdot \log n)$ 으로 가정한다. 이때  $n$ 은 정렬을 수행할 때 큐에 있는 객체의 개수이다. 반면에  $C_{adapt\_coordinate}$ 는 단순히 좌표의 각 차원에 덧셈 연산을 수행한 것이므로 단지 두 번의 덧셈 연산 비용이 필요하다. 그러므로 탐색하는 셀의 개수가 증가할수록 그 비용차이는 커지게 된다.

## 5. 성능 평가

본 논문에서 제안하는 거리 관계 패턴을 기반한  $k$ -최근접 질의 처리 기법의 특성을 보이기 위해 CPM 기법과의 성능 비교를 수행한다. 4장에서 도출된 수식과 성능에 영향을 주는 인자들을 이용하여 시뮬레이션을 해본 결과와 실제 알고리즘 구현에 의한 결과는 동일한 패턴을 나타내었다. 그래서 본 논문에서는 알고리즘 구현에 의한 결과들을 이용하여 성능을 평가한다.

성능 평가는 객체의 개수, 질의의 개수, 객체의 이동, 질의의 이동에 따른 질의 처리 속도를 평가하고 또한 저장 공간의 효율성에 대해 평가한다. 연속 질의 평가를 위해 객체의 속도, 이동 비율, 질의의 속도 그리고 이동 비율에 대해서 평가를 수행했다. 구현 및 수행은 IBM xSeries 336, CPU 3.2GHz, RAM 2Gbyte에서 이루어졌다. 제안하는 기법은 PB-kNN으로 표기한다. 표 3은 성능 평가에서 사용하는 인자들을 나타낸다.  $k$ -최근접 질의 처리 성능을 평가하기 위해  $N_{ctr}$ 의 크기는 1에서  $10^2$ 까지 변화시켜 보았다. 그리드 크기는  $64^2$ 에서  $1,024^2$ 까지, 객체 개수는 100K에서 1000K까지, 질의 개수는 1K에서 10K까지,  $k$ 값은 1에서 256까지 변화시키면서 평가를 수행하였다.

$k$ -최근접 질의 처리시 거리 관계 패턴이 필요로 하는 저장 공간의 크기와 질의 처리 성능의 관계를 조사하였

표 3 성능 평가 인자

구분	기본값	설정 범위
$N_{ctr}$	$4^2$	$1 \sim 10^2$
그리드 크기	$512^2$	$64^2 \sim 1024^2$
객체 개수	300K	100K~1000K
질의 개수	5K	1K~10K
$k$ 값	32	$1 \sim 256$

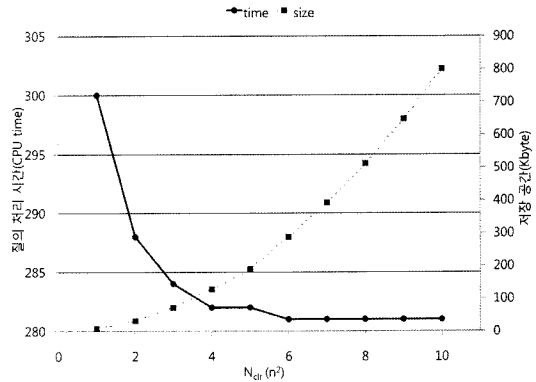


그림 8  $N_{ctr}$ 에 따른 저장 비용과 질의 처리 비용

다. 이것은 거리 관계 패턴이  $N_{ctr}$ 의 값에 따라 최대 오차  $E$ 가 결정되기 때문이다. 그림 8은  $N_{ctr}$ 의 증가에 따른 질의 처리 비용과 거리 관계 패턴이 차지하는 저장 공간의 관계를 보여준다.  $N_{ctr}$ 의 값이 4부터 질의 처리 비용은 거의 변화가 없지만 거리 관계 패턴의 크기는 여전히 증가하는 것을 볼 수 있다.

적절한 그리드의 크기를 결정하기 위해 그리드의 조밀도가 질의 처리를 위해 소모되는 연산 비용과 저장 비용에 미치는 영향을 조사하였다. 그림 9는 동일한 크기의 공간에 대해서 그리드 색인의 조밀도에 따른 질의 처리 속도의 변화를 나타낸다. 그리드의 크기에 관계없이 PB-kNN이 CPM보다 빠른 질의 처리 속도를 보장한다. 그리드 크기가 커질수록 두 알고리즘 모두 질의 처리 속도가 저하되지만, CPM의 속도 저하율 보다 PB-kNN의 속도 저하율이 작음을 나타낸다. 그리드의 조밀도가 커짐에 따라 CPM의 경우 근접한 셀을 찾기 위해 사용하는 우선순위 큐로 인해 비롯되는 연산 비용이 커지게 된다. 하지만 PB-kNN은 단순히 CLR을 순회하면서 질의를 처리하기 때문에 CPM보다는 그리드의 조밀도에 따른 영향을 적게 받는다. 그리드 크기가  $64^2 \sim 256^2$ 인 경우까지는 속도 차이가 15~25%정도로 증가하며 그 이후부터는 보다 큰 폭으로 벌어지는 것을 확인할 수 있다. 그리드 크기가 약  $256^2$ 인 지점에서 'U'형태로 처리율의 변화가 달라지게 된다. 한 셀당 포함하는 객체의 수가 증가하면 질의를 처리하기 위해 셀을 탐색



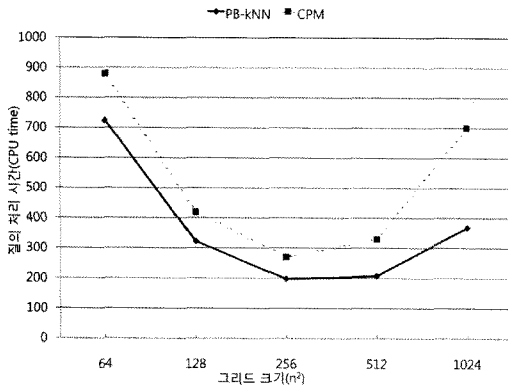


그림 9 그리드 크기에 따른 질의 처리 속도의 변화

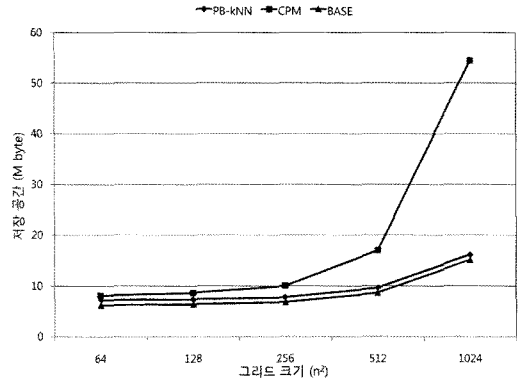


그림 10 그리드 크기에 따른 저장 비용

할 때마다 그 셀의 객체와 질의 점 간의 거리연산 비용도 증가하게 된다. 그래서 객체의 수에 비해 셀의 조밀도가 작은 경우 오히려 연산 비용이 증가하게 된다. 이러한 현상은 식 (30)에서 나타내고 있으며, 최적의 성능을 위해서는 객체의 수를 고려하여 적절한 조밀도로 그리드 크기를 결정해야 한다.

그림 10은 그리드 크기에 따른 저장 비용의 변화를 나타낸다. BASE는 질의 테이블과 거리 관계 패턴을 제외한 기본 저장 비용을 나타낸다. 다시 말해서 BASE는 객체 리스트와 질의 리스트를 포함하는 그리드 정보와 객체 테이블 정보를 유지하는데 요구되는 저장공간의 크기를 나타낸다. BASE에 해당하는 부분은 질의 처리 기법에 관계없이 연속 질의 처리를 위해 반드시 유지되어야 하는 정보이다. 나머지 자료구조는 질의 처리 기법의 특성에 따라 달라진다. CPM의 경우 질의 테이블에 우선순위 큐와 셀 방문 리스트를 유지하여 연속 질의를 처리한다. PB-CNN의 경우 CPM이 질의 테이블에 유지하는 우선순위 큐와 셀 방문 리스트를 유지하지 않고, 모든 질의가 공유해서 사용하는 거리 관계 패턴을 사용한다. PB-CNN은 모든 질의가 공유하는 거리 관계 패턴을 이용하기 때문에 셀의 조밀도에 따라 BASE에서 추가적으로 요구되는 저장공간이 일정하지만, CPM의 경우 조밀도가 커지면 방문하는 셀의 개수가 늘어나기 때문에 각 질의에서 유지하는 정보량이 크게 증가한다. 저장공간의 차이는 그리드 크기에 비례하여 증가하며, 그리드 크기가 64<sup>2</sup>에서 512<sup>2</sup>로 변화하는 동안 PB-kNN이 CPM에 비해 약 10%에서 35%의 저장공간 소모율의 감소를 보여준다. 1024<sup>2</sup>부터 그 차이는 60% 이상이 된다. 결론적으로 CPM이 소모하는 저장 비용에 비해 PB-kNN이 소모하는 저장비용은 매우 작고, 확장성이 있다는 것을 나타낸다.

그림 11은 객체의 개수에 따른 질의 처리 시간의 변

화를 나타낸다. 객체의 밀도가 증가하면  $k$ -최근접 객체를 찾기 위해 탐색해야 하는 셀의 개수가 감소하기 때문에 근접한 셀을 찾기 위한 연산이 감소한다. 그래서 객체수가 증가하면 질의 처리 시간이 감소한다. 이러한 현상은 CPM과 PB-kNN에서 모두 발생한다. 그림 11에서 객체 개수가 50K에서 500K개로 변화하는 동안, CPM의 질의 처리 시간은 약 2,700에서 270으로 감소하고 PB-kNN의 질의 처리 시간도 약 500에서 200으로 감소하는 것을 보여준다. 객체 수에 따른 PB-kNN의 질의 처리 속도는 CPM에 비해서 적게는 1.3배에서 많게는 5배 이상이 된다. 전체적으로 PB-kNN의 질의 처리 속도가 CPM보다 객체의 밀도에 관계없이 모든 상황에서 빠르다는 것을 확인시켜준다.

그림 12는 질의 개수에 따른 질의 처리 시간의 변화를 나타낸다. 질의 개수의 증가는 저장공간의 증가뿐만 아니라 질의 처리 시간의 증가도 초래한다. 질의 개수가 증가하기 때문에 발생하는 당연한 결과이다. 본 논문의 실험환경에서는 PB-kNN의 질의 개수에 상관없이 CPM

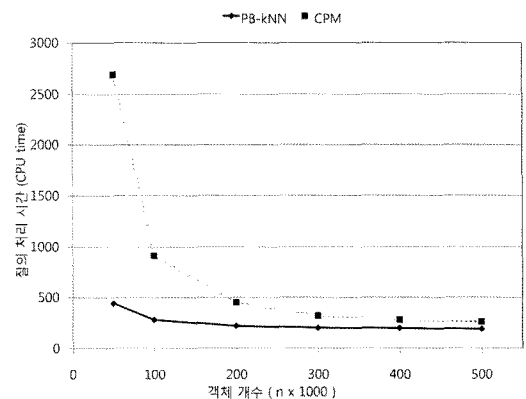


그림 11 객체 개수에 따른 질의 처리 시간 변화

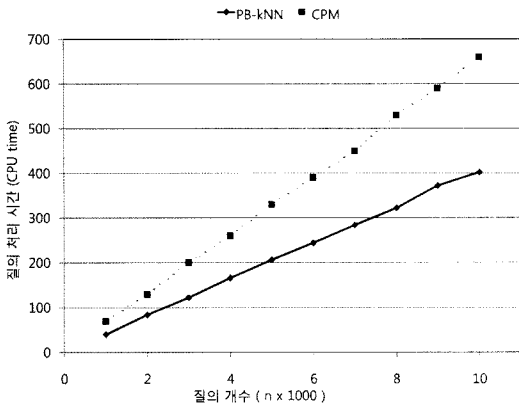


그림 12 질의 개수에 따른 질의 처리 시간 변화

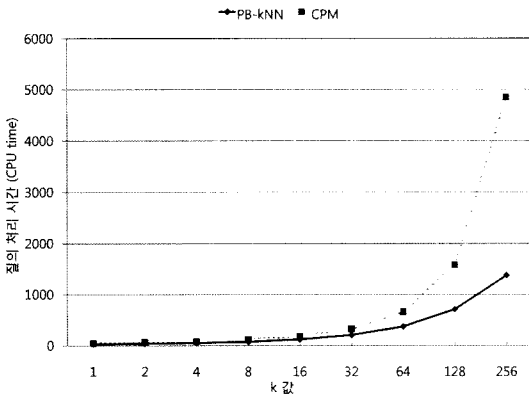


그림 13 k값에 따른 질의 처리 시간 변화

보다 적은 연산 비용을 소모하고, 질의 처리 시간을 약 30% 정도 단축하였다.

그림 13은 k값에 따른 질의 처리 시간의 변화를 나타낸다. k값을 1~256까지 변경하며 평가하였다. k값이 증가함에 따라 성능의 차이가 커짐을 발견할 수 있다. k값의 증가는 결국 탐색되는 셀의 개수를 증가시키기 때문이다. 두 기법 모두 k값이 작을 때는 거의 비슷한 시간 변화를 가지지만, 그 값이 커짐에 따라 시간 변화의 차이가 커지게 된다. 그림 13에서는 k값이 증가함에 따라 질의 처리 시간의 변화 비율은 CPM이 PB-kNN보다 약 4배 정도로 급격하게 변화되는 것을 보여준다. 결론적으로 PB-kNN이 CPM에 비해 k값의 증가에 따른 연산 비용의 증가율을 감소시켰다.

### 6. 결론

본 논문은 효율적인 연속 k-최근접 질의 처리를 위해 그리드 환경에서의 거리 관계 패턴을 제안하였다. 동일

한 모양과 크기를 갖는 셀들이 동일한 간격으로 배치되어 있을 경우 제안하는 거리 관계 패턴은 적은 연산 비용으로 근접한 셀들을 순서대로 탐색해 질의 처리 비용을 획기적으로 감소시켰고, 저장 비용의 낭비를 방지하였다. 또한 수식을 이용하여 PB-kNN과 CPM의 질의 처리 비용을 분석하였다. 마지막으로, 대표적인 k-최근접 질의 처리 기법인 CPM과의 성능 비교를 통해, 저장 비용과 처리 비용 측면에서 제안하는 k-최근접 질의 처리 기법의 성능이 모두 우수함을 확인하였다. 향후 연구로서 본 연구를 확장하여 거리 관계 패턴을 이용한 연속 Top-k 질의와 Skyline 질의 처리 기법을 연구할 것이고, 동시에 연속 질의 처리를 지원하는 효율적인 질의 색인 기법을 연구를 할 것이다.

### 참고 문헌

- [1] X. Yu, K. Pu, and N. Koudas, "Monitoring k-nearest Neighbor Queries over Moving Objects," *Proc. International Conference on Data Engineering*, pp. 631-642, 2005.
- [2] K. Mouratidis, M. Hadjieleftheriou, and D. Papadias, "Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring," *Proceedings of the ACM Conference on Management of Data*, pp. 634-645, 2005.
- [3] S. Prabhakar, Y. Xia, D. V. Kalashnikov, W. G. Aref and S. E. Hambrusch, "Query Indexing and Velocity Constrained Indexing : Scalable Techniques for Continuous Queries on Moving Objects," *IEEE Transactions on Computers*, Vol.51, No.10, pp. 1124-1140, 2002.
- [4] M. F. Mokbel, X. Xiong, and W. G. Aref, "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases," *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 623-634, 2004.
- [5] X. Wang, Q. Zhang, W. Sun, W. Wang, and B. Shi, "cGridx: Efficient Processing of Continuous Range Queries over Moving Objects," *Proc. International Conference on Advances in Web-Age Information Management*, pp. 345-356, 2005.
- [6] D. V. Kalashnikov, S. Prabhakar, and S. E. Hambrusch, "Main Memory Evaluation of Monitoring Queries over Moving Objects," *Distributed and Parallel Databases*, Vol.15, No.2, pp. 117-135, 2004.
- [7] K. L. Wu, S. K. Chen, and P. S. Yu, "Processing continual range queries over moving objects using VCR-based query indexes," *Proc. IEEE International Conference on Mobile and Ubiquitous Systems : Networking and Services*, pp. 226-235, 2004.
- [8] K. L. Wu, S. K. Chen, and P. S. Yu, "On Incremental Processing of Continual Range Queries for Location-Aware Services and Applications," *Proc.*

*Annual International Conference on Mobile and Ubiquitous Systems*, pp. 261-269, 2005.

- [9] K. Mouratidis and D. Papadias, "Continuous Nearest Neighbor Monitoring in Road networks," *Proc. the Very Large data bases Conference (VLDB)*, 2006.
- [10] K. Mouratidis and D. Papadias, "Continuous Nearest Neighbor Queries over Sliding Windows," *IEEE Transactions on Knowledge and data Engineering(TKDE)*, Vol.19, No.6, pp. 789-803, 2007.

박 용 훈

정보과학회논문지 : 데이터베이스  
제 36 권 제 2 호 참조

서 동 민

정보과학회논문지 : 데이터베이스  
제 36 권 제 2 호 참조



박 경 수

1998년 충북대학교 수학과(이학사). 2000년 충북대학교 정보통신공학과(공학석사) 2005년 충북대학교 정보통신공학과(공학박사). 2005년 3월~2008년 2월 한국과학기술원 정보전자연구소 연수연구원. 2008년 3월~현재 (주)가인정보기술 차장. 관심분야는 자료 저장 시스템, 이동객체 데이터베이스, 시공간 색인구조, 센서 네트워크 및 RFID 등



이 병 엽

1991년 2월 한국과학기술원 전산학과(공학사). 1993년 2월 한국과학기술원 전산학과(공학석사). 1997년 2월 한국과학기술원 경영정보공학(공학박사). 1993년 1월~2003년 2월 대우정보시스템 차장 2003년 3월~현재 배재대학교 전자상거래학과 부교수. 관심분야는 XML, 지능정보시스템, 데이터베이스 시스템, 전자상거래학

유 재 수

정보과학회논문지 : 데이터베이스  
제 36 권 제 1 호 참조