韓國컴퓨터情報學會 論文誌
第15卷 第1號, 2010. 1.

2010-15-1-1-9

# 순환검색거리를 이용하는 최대근접 질의처리의 성능분석

선 휘 준*, 김 원 호**

# The Performance Analysis of Nearest Neighbor Query Process using Circular Search Distance

Hwi-Joon Seon*, Won-Ho Kim**

## 요 약

최대근접질의의 처리비용을 최적화하기 위해서는 색인에서 검색되는 노드의 수와 연산시간을 최소화할 수 있어야 한다. 이를 위해 최대근접질의 처리시 검색대상을 정확히 선택하고 객체들의 순환적 위치 속성이 고려된 검색거리 측도가 필요하다. 본 논문은 순환도메인을 갖는 검색공간에서 객체의 순환적 위치속성을 고려한 최대근접질의 처리방법을 제안하고 그 성능을 실험을 통하여 입증한다. 제안한 방법은 최대근접질의의 처리비용을 최적화하기 위한 검색거리 측도인 순환최소거리와 순환최적거리를 사용한다.

## Abstract

The number of searched nodes and the computation time in an index should be minimized for optimizing the processing cost of the nearest neighbor query. The Measurement of search distance considered a circular location property of objects is required to accurately select the nodes which will be searched in the nearest neighbor query. In this paper, we propose the processing method of the nearest neighbor query be considered a circular location property of object where the search space consists of a circular domain and show its performance by experiments. The proposed method uses the circular minimum distance and the circular optimal distance which are the search measurements for optimizing the processing cost of the nearest neighbor query.

▸ Keyword : 순환위치속성(Circular Location Property), 순환최소거리(Circular Minimum Distance), 순환최적거리(Circular Optimal Distance), 최대근접질의(Nearest Neighbor Query)

• 제1저자 : 선휘준
• 투고일 : 2009. 11. 02, 심사일 : 2009. 12. 21, 게재확정일 : 2010. 01. 26.
* 신경대학교 애니메이션학과 교수 ** 신경대학교 인터넷정보통신학과 교수

# Ⅰ. 서 론

최근의 정보 서비스들은 대용량의 멀티미디어 시스템을 기반으로 하고 있다. 이러한 멀티미디어 시스템에서는 주어진 위치에서 가장 가까운 객체를 찾는 최대근접질의가 자주 발생한다[1,2,4,6,7,8]. 그러나 적중 에러 없이 정확한 최대근접객체를 검색하기 위해서는 연산 및 보조기억장치 접근을 위한 많은 처리 시간이 요구된다.

기존의 최대근접질의 처리방법들은 객체의 속성이 가질 수 있는 최소와 최대값 사이의 선형적 순서 범위로 구성된 검색공간을 가정하였다. 그러나 지리정보시스템과 같은 응용에서 취급되는 객체 또는 시간의 개념이 포함되어 있는 객체 등은 순환적인 위치속성을 가질 수 있다. 따라서 최대근접질의의 처리시 이러한 객체의 속성을 효율적으로 반영하기 위해서는 순환적인 성질을 갖는 도메인으로 구성된 검색공간과 새로운 검색거리 측도가 필요하다.

본 논문에서는 순환도메인을 갖는 검색공간에서 객체의 순환적 위치 속성이 고려된 최대근접질의의 처리방법을 제안한다. 그리고 최대근접질의의 처리비용을 최적화하기 위한 검색거리 측도인 순환최소거리와 순환최적거리를 정의한다. 또한 제시된 검색거리 측도를 R*-트리[5]에 적용하여 기존의 방법과 최대근접질의에 따른 처리 비용을 비교 평가한다. 질의 처리 비용을 비교 평가하기 위한 지수로는 디스크 접근 횟수, 질의처리시간을 이용하였으며, 객체들이 검색공간에 균일하게 발생되는 경우와 가장자리에 집중되어 발생하는 경우 그리고 순환적인 특성이 많이 적용되어 일부분에 집중되어 발생되는 경우 최대근접질의의 처리 성능을 규명하였다.

논문의 구성은 다음과 같다. 2장에서는 관련연구로서 기존의 최대근접질의 처리방법을 알아보고, 3장에서는 논의대상이 되는 검색거리 측도인 순환최소거리와 순환최적거리를 정의하고 이를 적용한 최대근접질의 알고리즘을 기술한다. 4장에서는 실험을 통하여 순환최소거리와 순환최적거리를 적용한 최대근접질의의 처리 성능을 기존의 방법과 비교 분석하고, 끝으로 5장에서는 결론을 내린다.

# Ⅱ. 관련연구

최대근접질의의 처리비용을 최적화하기 위해서는 연산 시간과 색인에서 검색되는 노드의 수를 최소화할 수 있어야 한다. 이를 위해 최대근접질의의 처리시 색인에서 방문될 노드들이 정확히 선정되도록 위치 속성에 의한 검색거리 측도인 최적탐색거리가 제안되었다[9]. 최적탐색거리는 질의기준으로부터 객체 또는 부검색공간들이 반드시 존재하는 거리 중에서 최소의 거리이며, 최대근접질의의 처리시 질의기준의 유형에 관계없이 색인에서 검색될 노드들을 정확히 선택하기 위한 검색거리 측도이다. [9]의 방법은 최대근접질의의 처리시 색인에서 검색대상이 되는 노드들의 수를 최적화하였으나, 선형적 순서 범위로 구성된 검색공간만을 전제로 하였다.

그러나 많은 응용에서는 시간의 개념이 포함된 최대근접질의가 자주 발생하며, 이러한 최대근접질의의 처리가 요구될 경우 객체들의 순환적 위치속성을 고려해야 한다.

따라서 최대근접질의 처리시 객체의 순환적 위치속성을 효율적으로 반영하기 위해서는 순환적 순서범위를 갖는 도메인으로 구성된 검색공간과 검색거리 측도가 요구된다.

[3]에서는 순환적 순서범위를 갖는 순환 도메인(circular domain)을 다음과 같이 정의하였다.

객체의 위치속성 값을 매개변수로 하여 임의 위치속성 값의 다음 또는 이전 위치속성 값을 반환하는 함수를 각각 $next$ , $prev$ 라 하자. $N$차원 검색공간을 구성하는 임의의 도메인 $D_i$ $(i=1,2,...,N)$가 다음과 같은 성질을 만족할 때, $D_i = \{d_1, d_2, ..., d_m\}$

(여기에서 $d_j < d_{j+1}$, $j=1,2,...,m-1$)를 순환 도메인이라 한다.

- $next(d_k) = d_{k+1} (1 \le k < m)$ 그리고 $next(d_m) = d_1$
- $prev(d_k) = d_{k-1} (1 < k \le m)$ 그리고 $prev(d_1) = d_m$

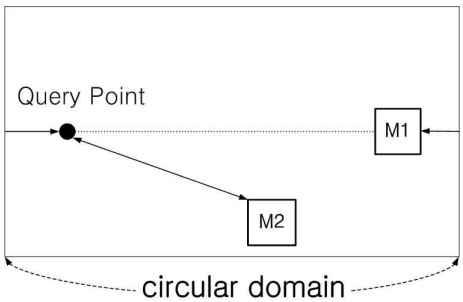



그림 1. 순환적인 검색공간
Fig. 1. Circular Search Space

그림 1은 $x$ 축이 순환적 순서범위를 갖는 도메인으로 구성된 검색공간에서 질의기준으로부터 가장 가까운 거리에 있는

객체를 선택하는 예이다. $DIST_p$를 질의기준으로부터 객체 $p$의 위치속성에 따른 유클리드 거리하고, $DIST'_p$를 순환적 범위를 갖는 도메인으로 구성된 검색공간에서 질의기준으로부터 객체 $p$까지의 최소 유클리드 거리라 하자. 질의기준으로부터 가장 가까운 객체를 찾는다고 할 때, 만약 순환적 순서범위를 고려하지 않고 유클리드 거리를 측도로 하여 대상을 선택한다면 $DIST_{M1} > DIST_{M2}$이 됨으로써 객체 $M2$을 최대근접객체로 선정하게 된다. 그러나 순환적 순서범위를 고려한 최소 유클리드 거리는 $DIST'_{M1} < DIST_{M2}$이 되기 때문에 최대근접객체는 $M1$이 된다.

## III. 순환최소거리와 순환최적거리를 이용한 최대근접질의 처리 방법

본 장에서는 순환도메인을 갖는 $N$차원 검색공간에서 최대근접질의의 처리를 위한 순환최소거리와 순환최적거리를 정의하고 이를 적용한 최대근접질의 알고리즘을 기술한다. $N$차원 검색공간에서 질의기준이 되는 $Q$는 범위들의 카테시언 곱(cartesian product)으로 표현된다. 즉, 도메인 $D_i$에 발생하는 범위를 $I_i(Q) = [Q_{Li}, Q_{Ui}](i = 1,\cdots,N,\ Q_{Li}, Q_{Ui}$ : 범위의 시작과 끝)라 할 때, $Q = I_1(Q) \times I_2(Q) \times \cdots \times I_N(Q)$ 로 나타낼 수 있다.

본 논문에서는 순환도메인이 포함된 $N$차원 검색공간에서 질의기준 $Q$와 최소경계사각형 $M$사이의 가장 가까운 거리인 순환최소거리(Circular MINimum DISTance: CMINDIST)를 다음과 같이 정의한다.

【정의 1】 $D_i = \{d_1, d_2, ..., d_m\}$ (여기에서 $d_j < d_{j+1}$, $j = 1,2,\cdots,m-1$)를 $N$차원 검색공간을 구성하는 $i$번째 도메인이라 할 때, $N$차원 검색공간에서 질의기준 $Q$와 최소경계사각형 $M$간의 순환최소거리 $CMINDIST$는 다음과 같다.

$$CMINDIST(Q,M) = \sum_{i=1}^{N} d_i^2(Q,M)$$

만약 $D_i$가 순환도메인이면

$d_i(Q,M) = \min(|Q_i - M_i|, |q_i - d_1| + |m_i - d_m|)$

그렇지 않으면

$d_i(Q,M) = |Q_i - M_i|$

여기에서

$$Q_i = \begin{cases} Q_{Li}, & Q_{Li} > M_{Ui} \\ Q_{Ui}, & Q_{Ui} < M_{Li} \\ 0, & otherwise \end{cases}$$

$$M_i = \begin{cases} M_{Li}, & Q_{Ui} < M_{Li} \\ M_{Ui}, & Q_{Li} > M_{Ui} \\ 0, & otherwise \end{cases}$$

$$q_i = \begin{cases} Q_{Li}, & Q_{Li} > M_{Ui} \\ Q_{Li}, & Q_{Ui} < M_{Li} \\ 0, & otherwise \end{cases}$$

$$m_i = \begin{cases} M_{Ui}, & Q_{Ui} < M_{Li} \\ M_{Ui}, & Q_{Li} > M_{Ui} \\ 0, & otherwise \end{cases}$$

$d_i(Q,M)$: $D_i$에서 $Q$와 $M$사이의 거리.

정의된 $CMINDIST$는 순환도메인이 포함된 $N$차원 검색공간에서 최소경계사각형 $M$에 포함되어 있는 부검색공간들 중에서 질의기준 $Q$에 가장 근접하고 있는 객체 또는 부검색공간을 결정하기 위한 최소의 거리이다.

색인의 검색 시 불필요한 노드의 검사를 피하고 다음 검색대상이 되는 노드의 수를 최소로 하는 검색거리 측도가 필요하다. 이를 위해 본 논문에서는 객체들의 순환적 위치속성이 반영된 순환최적거리(Circular Optimized MINium value of all DISTances: COMINDIST)를 다음과 같이 제안한다. 제안될 순환최적거리는 질의기준 $Q$에서 최소경계사각형 $M$의 임의의 한 변을 포함할 수 있는 거리들 중 최소거리로 계산된다. 또한 순환도메인이 존재한다면 COMINDIST는 질의결과의 산출시 객체들의 순환적 위치속성을 정확히 반영한 새로운 거리측도이다.

【정의 2】 $N$차원 검색공간에서 질의기준 $Q$와 최소경계사각형 $M$간의 순환최적거리 $COMINDIST$는 다음과 같다.

$COMINDIST(Q,M) = \min\{d_i(Q,M), d_k(Q,M), d(Q,M)\}$

만약 $D_i$가 순환도메인이면

$d_i(Q,M) =$

$$\min_{1 \le i \le N} \left\{ \begin{array}{l} (|q_i - d_1| + |m_i - d_m|^2 + \sum_{\substack{i \ne k \\ 1 \le k \le N}} |Qr_k - Mr_k|^2), \\ (|q_i - d_1| + |M_i - d_m|^2 + \sum_{\substack{i \ne k \\ 1 \le k \le N}} |Qr_k - mr_k|^2) \end{array} \right\}$$

만약 $D_k$가 순환도메인이면

$d_k(Q,M) =$

$$\min_{1 \leq i \leq N} \left\{ \begin{array}{l} ( \mid qr_i - mr_i \mid^2 \\ \quad + \sum_{\substack{i \neq k \\ 1 \leq k \leq N}} \mid Q_k - d_1 \mid + \mid M_k - d_m \mid^2 ), \\ ( \mid qr_i - Mr_i \mid^2 \\ \quad + \sum_{\substack{i \neq k \\ 1 \leq k \leq N}} \mid Q_k - d_1 \mid + \mid m_k - d_m \mid^2 ) \end{array} \right\}$$

그렇지 않으면

$$d(Q, M) =$$

$$\min_{1 \leq i \leq N} \left\{ \begin{array}{l} ( \mid qr_i - mr_i \mid^2 \\ \quad + \sum_{\substack{i \neq k \\ 1 \leq k \leq N}} \mid Qr_k - Mr_k \mid^2 ), \\ ( \mid qr_i - Mr_i \mid^2 \\ \quad + \sum_{\substack{i \neq k \\ 1 \leq k \leq N}} \mid Qr_k - mr_k \mid^2 ) \end{array} \right\}$$

여기에서

$$qr_i = \begin{cases} Q_{Li}, & Q_{Li} \geq M_{Ui} \\ Q_{Ui}, & Q_{Ui} \leq M_{Li} \\ Q_i, & otherwise \end{cases}$$

$$Q_i = \begin{cases} Q_{Li}, & \frac{(Q_{Li} + Q_{Ui})}{2} \leq \frac{(M_{Li} + M_{Ui})}{2} \\ Q_{Ui}, & otherwise \end{cases}$$

$$q_i = \begin{cases} Q_{Li}, & Q_{Li} \geq M_{Ui} \\ Q_{Li}, & Q_{Ui} \leq M_{Li} \\ Q_{Ui}, & otherwise \end{cases}$$

$$Qr_k = \begin{cases} Q_{Lk}, & \frac{(Q_{Lk} + Q_{Uk})}{2} \geq \frac{(M_{Lk} + M_{Uk})}{2} \\ Q_{Uk}, & otherwise \end{cases}$$

$$mr_i = \begin{cases} M_{Li}, & \frac{(Q_{Li} + Q_{Ui})}{2} \leq \frac{(M_{Li} + M_{Ui})}{2} \\ M_{Ui}, & otherwise \end{cases}$$

$$m_i = \begin{cases} M_{Ui}, & \text{if} \frac{(Q_{Li} + Q_{Ui})}{2} \leq \frac{(M_{Li} + M_{Ui})}{2} \\ M_{Ui}, & otherwise \end{cases}$$

$$Mr_k = \begin{cases} M_{Lk}, & \text{if} \frac{(Q_{Lk} + Q_{Uk})}{2} \geq \frac{(M_{Lk} + M_{Uk})}{2} \\ M_{Uk}, & otherwise \end{cases}$$

$$M_k = \begin{cases} M_{Lk}, & \text{if} \frac{(Q_{Lk} + Q_{Uk})}{2} \geq \frac{(M_{Lk} + M_{Uk})}{2} \\ M_{Lk}, & otherwise \end{cases}$$

정의된 COMINDIST는 질의기준$Q$와 최소경계사각형$M$간에 겹치지 않는 경우만을 고려하였으나, $N$차원 검색공간에서 $Q$와 $M$이 겹쳐있는 경우로 확장하여 정의될 수 있다. 순환도메인이 포함된 $N$차원 검색공간에서 최대근접질의의 처리가 요구될 경우 COMINDIST와 CMINDIST를 이용하면 검색공간에 존재하는 객체들에 대한 검색범위를 가장 작게 유지하므로 색인에서 검색되는 노드의 수를 줄인다.

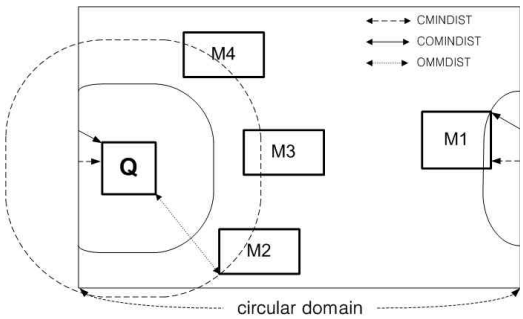



그림 2. 순환적인 검색공간에서 COMINDIST와 OMMDIST
Fig. 2. COMINDIST and OMMDIST in Circular Search Space

그림 2는 $x$ 축이 순환적 순서범위를 갖는 도메인으로 구성된 검색공간에서 [9]에서 제시한 OMMDIST와 본 논문에서 정의한 COMINDIST의 검색범위를 비교한 것이다. 예에서는 OMMDIST에 의한 방법은 M2, M3, M4에 해당하는 노드가 검색대상이 되고, COMINDIST에 의한 방법은 M1만이 다음 검색대상이 된다. 따라서 객체들의 순환적 위치속성이 반영된 COMINDIST와 CMINDIST를 이용하면 필요하지 않는 3개의 노드 접근을 줄일 수 있다.

순환최적거리를 이용하면 순환적인 검색공간에 존재하는 객체들에 대한 검색범위를 가장 작게 유지하므로 색인에서 검색되는 노드의 수를 줄인다. 따라서 최대근접질의 처리 시 보조기억장치의 접근횟수를 최소화할 수 있다.

본 논문에서 제안하는 최대근접질의의 처리 방법은 색인을 검색하는 동안 방문할 필요가 없는 노드들을 방문대상에서 제외하기 위해 다음과 같은 전략을 사용한다.

ⅰ) 질의 기준 Q로부터 최소경계사각형 M′까지의 COMINDIST(Q,M′)보다 CMINDIST(Q,M)가 더 큰 값을 갖는 최소경계사각형 M이 존재하면, M은 최대근접객체를 포함하지 않기 때문에 M에 해당하는 노드는 검색대상에서 제외한다.

ⅱ) 질의 기준 Q로부터 객체 O까지의 거리가 최소경계사각형 M에 대한 COMINDIST(Q,M)보다 크다면, 객체 O를 최대근접객체 대상에서 제외한다.

ⅲ) 질의 기준 Q로부터 객체 O까지의 거리보다 더 큰 CMINDIST(Q,M)를 갖는 모든 최소경계사각형 M은 검색대상에서 제외한다.

본 논문에서 제안하는 최대근접질의 알고리즘은 주어진 질

의기준으로부터 가장 가까운 객체를 찾기 위한 검색연산이 루트 노드부터 시작해서 하위 레벨의 노드까지 재귀적인 방법으로 이루어진다.

```
Procedure NEAREST_NEIGHBOR_SEARCH
BEGIN
  IF leaf_node(NODE) THEN
    FOR all entries of NODE DO
      sort_entry(QueryPoint,NODE.Entry);
      List1:=
          Prunning_Node_Entry(QueryPoint,NODE.Entry);
      FOR all List1 DO
        Get_Bucket(NODE.Entry,BUCKET);
        FOR all objects of BUCKET DO
          sort_entry(QueryPoint,BUCKET.Object);
          List2:=
            Prunning_Object(QueryPoint,BUCKET.Object);
          FOR all List2 DO
            dist:=object_dist(QueryPoint,BUCKET.Object);
            IF dist < NEAREST.dist THEN
                NEAREST.dist:=dist;
                NEAREST.object:=BUCKET.Object;
            ENDIF
          ENDFOR
        ENDFOR
      ENDFOR
    ENDFOR
  ELSE
    FOR all entries of NODE DO
        sort_entry(QueryPoint,NODE.Entry);
        Prunning_Node_Entry(QueryPoint,NODE.Entry);
        NEAREST_NEIGHBOR_SEARCH;
    ENDFOR
  ENDIF
END
```

그림 3. 순환검색거리에 의한 최대근접질의 알고리즘
Fig. 3. Nearest Neighbor Query Algorithm according to Circular Search Distance

그림 3의 알고리즘에서 sort_entry는 정의 1에 의해 구해진 질의기준과 노드의 각 엔트리간의 CMINDIST를 오름차순으로 순서화하는 함수이다. Prunning_Node_Entry는 CMINDIST에 의해 순서화된 엔트리들을 정의 2에 의해 구해진 COMINDIST와 비교 연산후 검색할 필요가 없는 노드들을 제거하는 절차이다. Prunning_Object는 버켓에 있는 각각의 객체들에 대해서 질의기준으로부터 객체까지의 거리보다 먼 CMINDIST인 최소경계사각형들을 제거하는 절차이다. 또는 질의기준으로부터 객체까지의 거리가 최소경계사각형까지의 COMINDIST보다 크다면, 객체를 최대근접객체 대상에서 제외하는 절차이다. 그리고 object_dist는 질의 기준과 버켓에 있는 객체들 사이의 거리를 계산하는 함수이다.

## IV. 실험을 통한 성능 평가

본 장에서는 순환최적검색거리 COMINDIST의 성능을 질의기준으로부터 가장 가까운 검색대상 노드들을 찾는데 소요되는 디스크 접근 횟수와 질의처리시간에 따라 평가하였다. 실험에서는 R*-트리에 순환최적검색거리 COMINDIST와 최적탐색거리 OMMDIST[9]에 의한 최대근접질의 처리 알고리즘을 적용한 후 이에 따른 실험결과에 의해 그 성능을 비교 평가하였다.

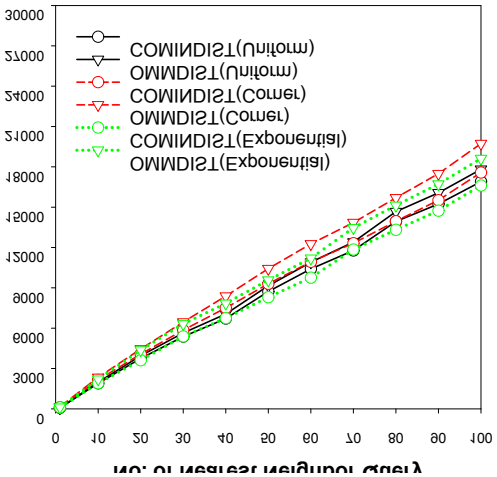
실험에서는 이차원 검색공간에서 중복되지 않은 30,000개의 사각형 객체를 사용하였으며, 논문에서 제안한 검색거리 측도의 성능을 명확히 보이기 위해 동일한 비율 및 80바이트의 고정된 크기를 갖는 사각형 객체를 가정하였다. 또한 객체들의 면적은 전체 검색공간의 0.1%, 0.001%인 경우를 고려하였다. 이는 색인에 삽입되는 대부분의 객체가 어느 정도 겹치는 경우(0.1%)와 거의 겹치지 않는 경우(0.001%)에 대해서 검색거리 측도의 특성을 보이기 위해서이다.

하나의 도메인의 범위가 [0,1)이라할 때, 실험에서 사용된 객체의 분포는 다음과 같다.

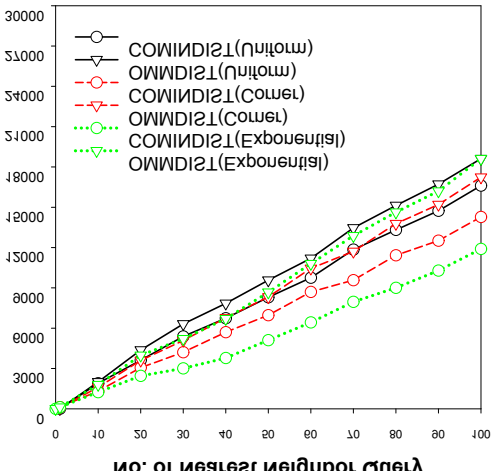
- 균일분포(uniform distribution) : 객체들의 중심점이 중복되지 않고 균일하게 분포
- 모서리분포(corner distribution) : 객체들의 중심점이 평균 0.5인 이차 함수 분포
- 지수분포(exponential distribution) : 객체들의 중심점이 평균 0.5인 지수 함수 분포

균일분포는 객체의 분포가 이상적인 상태에서의 최대근접질의의 처리성능을 규명하기 위한 것이다. 그리고 모서리 분포는 객체들이 검색공간의 가장자리에 집중되어 발생하는 경우, 지수분포는 객체들의 순환적 위치 속성을 고려하여 최대근접객체를 찾아야 하는 경우에 최대근접질의의 처리성능을 규명하기 위한 것이다. 디스크에 저장된 R*-트리의 노드와 버켓은 동일한 크기를 가지며, 모든 노드와 버켓의 접근시간은 동일하다고 가정하였다.

실험에서는 100개의 최대근접질의를 균일하게 발생시켜 디스크 접근 횟수와 질의처리시간에 따른 성능을 알아보았다.




(a) 질의크기=0.01%

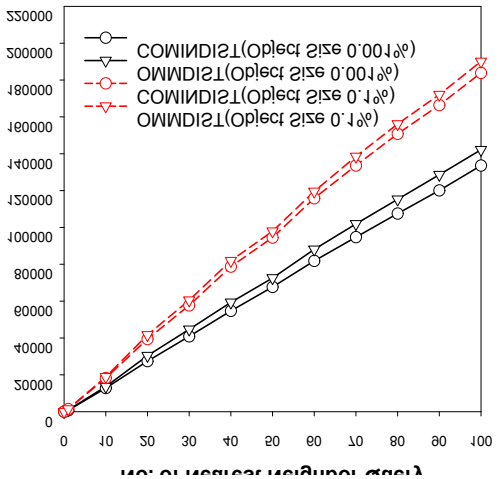



(b) 질의크기=1.0%

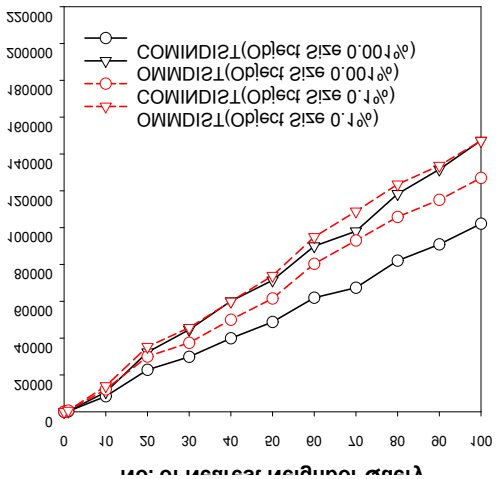
그림 4. 질의크기에 따른 디스크접근횟수
Fig. 4. The Number of Disk Access according to Query Size

질의의 크기는 전체 검색공간의 넓이를 $S$ 라고 했을 때, $S\times(0.1)^{n+1}\times c$의 넓이를 가지는 사각형 영역이다. 예를 들어 $n=1$이고 $c=1$이면, 질의 크기는 전체 검색공간의 1.0%되는 사각형 영역이 된다. 실험에서는 순환검색거리의 특성을 명확히 규명하기 위해서 질의 크기를 전체 검색공간의 0.01%, 1.0%로 하였다.

그림 4는 균일분포, 모서리분포 그리고 지수분포에서 객체의 크기가 0.1%일 때 질의 수의 증가에 따른 디스크 접근 횟수를 나타낸 것이다. 그림에서는 질의크기에 관계없이 질의의 개수가 많아질수록 디스크 접근 횟수가 선형적인 증가를 보이며, COMINDIST와 OMMDIST의 성능 차이가 더 커짐을 보인다. 또한 질의의 크기가 크고, 객체들이 검색공간의 가장자리의 일부분에 집중되고 객체들의 순환적인 위치속성을 고려해야 하는 분포일수록 그 차이가 더 커짐을 알 수 있다.




(a) 균일분포




(b) 지수분포

그림 5. 객체크기에 따른 질의처리시간
Fig. 5. Query Processing Time according to Object Size

반복된 실험결과에 의하면 COMINDIST를 이용한 처리 방법은 질의 크기나 객체의 분포에 상관없이 OMMDIST에 의한 처리 방법보다 항상 낮은 디스크 접근 횟수를 보였다. 그리고 각각의 분포에서 질의의 크기가 1.0%일 때에는 COMINDIST와 OMMDIST의 디스크 접근 횟수의 차가 더 커짐을 알 수 있었다.

이러한 결과는 순환적인 특성이 많이 적용된 분포일수록 색인에서 검색되는 노드의 선택 시 COMINDIST에 의한 검색거리가 OMMDIST에 의한 검색거리보다 더 작기 때문이다. 따라서 검색대상에서 제외되는 노드 및 버켓들이 OMMDIST에 의한 방법보다 COMINDIST에 의한 방법이 더 많기 때문에 결과적으로 디스크 접근 횟수가 더 적어지기 때문이다.

그림 5는 최대근접질의에 따른 질의 처리 시간을 분석하기 위해 객체의 크기가 0.001%, 0.1%이고 질의 크기가 1.0%일 때 100개의 최대근접질의를 처리하는데 따른 누적된 질의 처리 시간을 나타낸 것이다.

그림에서는 객체들의 순환적 특성이 거의 적용되지 않은 균일분포의 경우 객체의 크기에 관계없이 COMINDIST를 이용한 처리 방법과 OMMDIST를 이용한 처리 방법과의 질의 처리 시간은 비슷한 경향을 보여준다. 그러나 객체들이 검색공간에서 거의 겹치지 않고 한 영역에 집중적으로 발생하며 순환적인 특성이 많이 적용된 지수분포의 경우 COMINDIST를 이용한 처리 방법과 OMMDIST를 이용한 처리 방법과의 질의 처리 시간의 차가 더 커짐을 보여준다. 이러한 이유는 색인에서 방문대상이 되는 노드들의 선정 시 OMMDIST에 의한 방법이 COMINDIST에 의한 방법보다 더 많은 노드들을 선택하기 때문이다. 그러므로 질의의 수가 많아질수록 검색되는 노드의 수가 가중되어 이에 따른 질의 처리 시간이 더욱 커지게 된다.

반복된 실험결과에 의하면 객체들의 모든 분포에 있어서 대부분의 객체가 겹치는 경우인 0.1%일 때에는 COMINDIST를 이용한 방법과 OMMDIST를 이용한 방법의 질의 처리 시간은 비슷한 경향을 보임을 알 수 있었다.

## V. 결 론

대용량의 멀티미디어 시스템에서 최대근접객체를 찾는 질의의 처리는 많은 디스크 접근과 질의처리시간을 요구한다. 또한 데이터의 차원이 증가함에 따라 검색비용이 크게 증가할 수 있다. 따라서 최대근접질의의 처리비용을 최적화하기 위해서는 색인에서 검색되는 노드의 수를 최소화할 수 있는 검색거리 측도가 필요하다.

본 논문에서는 N차원 검색공간에서 객체들의 순환적 위치 속성을 고려한 검색거리측도인 순환최소거리와 순환최적거리를 제시하고 이를 이용한 최대근접질의의 처리 방법을 제안하였다. 그리고 실험을 통하여 기존의 방법과 질의처리 성능을 비교 분석하였다.

실험에서는 최대근접질의의 성능을 디스크 접근 횟수와 질의처리시간에 따라 비교 평가하였다. 실험데이터로는 균일분포, 모서리분포 그리고 지수분포를 이루는 사각형 객체들을 이용하였다. 실험결과에 의하면, 순환최소거리와 순환최적거리를 이용한 최대근접질의의 처리는 객체의 분포형태, 객체의 크기, 질의크기에 관계없이 항상 낮은 디스크 접근횟수를 보였다. 특히 순환적인 특성이 많이 적용되는 분포일수록 순환최적검색거리를 이용한 최대근접질의의 처리가 최적탐색거리를 이용한 처리에 비해 디스크 접근 횟수가 더욱 작아짐을 보였다. 또한 질의의 크기가 크고 객체의 크기가 클수록 질의처리 시간이 상대적으로 더 적어짐을 알 수 있었다.

## 참고문헌

[1] C.Bohm, et al., "Searching in High-dimensional Spaces: Index Structures for Improving the Performance of Multimedia Databases," ACM Computing Surveys, Vol. 33, 2001.

[2] C.Yang, et al., "An Index Structure for Efficient Reverse Nearest Neighbor Queries," 17th Int. Conf. on Data Engineering(ICDE'01), pp. 485-493, 2001.

[3] H.J.Seon, et al., "The Spatial Indexing Method for Supporting a Circular Location Property of Object," Proc. AIRS, pp.147-159, 2005.

[4] K.L.Cheung, et al.,"Enhanced Nearest Neighbor Search on the R-tree," IN Proc. ACM SIGMOD RECORD, Vol. 27, No. 3, pp.16-21, 1998.

[5] N.Beckmann, H.Kriegel, R.Schneider and B.Seeger, "The R*-tree: an Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 322-331, 1990.

[6] N.Roussopoulos, et al., "Nearest Neighbor Queries,"In Proc. Intl. Conf. on Management of Data, ACM SIGMOD, pp. 71-79, 1995.

[7] Y.Tao, et al., "Continuous Nearest Neighbor Search," Proc. the 28th VLDB Conf., pp. 287-298, 2002.

[8] Y.Gao, et al., "Continuous Obstructed Nearest Neighbor Queries in Spatial Databases," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 577-590, 2009.

[9] 선휘준 외 1, "최적탐색거리를 이용한 최근접질의의 처리 방법의 성능 평가," 한국정보처리학회 논문지, 제 6권, 제1호, 32-41쪽, 1999년 1월.

## 저 자 소 개


**선 휘 준**

1990 : 전남대학교 이학석사.
1998 : 전남대학교 이학박사.
1997~2005 : 서남대학교 컴퓨터정보통신학과 교수
2006~현재 : 신경대학교 애니메이션학과 교수
관심분야 : 영상처리, 애니메이션,디지털콘텐츠, 공간자료처리

**김 원 호**

1996 : 전남대학교 교육학석사.
2001 : 전남대학교 박사수료.
1992~2005 : 서남대학교 물리학과 교수
2006~현재 : 신경대학교 인터넷정보통신학과 교수
관심분야 : 이산이론, 알고리즘, 그래픽스