

# 공간 질의의 시공간적 연관성을 이용한 버퍼관리기법

이정민<sup>o</sup>, 강명아, 이기준  
부산대학교 전자계산학과

e-mail: {jmlee, makang, lik}@spatios.cs.pusan.ac.kr

## Buffer Management using Spatial and Temporal Locality of Spatial Query

Jeong-Min Lee<sup>o</sup>, Myeong-A Kang and Ki-Joune Li  
Dept. of Computer Science  
Pusan National University

### 요약

지리정보시스템에서의 공간 질의는 시간적 공간적 집중성을 가지고 주어진다. 일반적으로 대용량의 데이터를 다루기 때문에 지리정보 시스템에서 요구하는 질의의 처리 효율은 중요한 문제가 된다. 본 논문에서는 공간 질의의 시간적 공간적 집중성을 고려하여 공간 질의의 처리 효율을 높일 수 있는 방법을 연구한다. 이 방법은 일련의 질의 결과로 탐색된 객체들을 지역적 특징을 이용하면서 버퍼링하여 다음 질의에 사용할 수 있도록 한다. 이러한 버퍼링은 검색 공간을 줄여서 디스크 참조를 적게 하고 공간 질의 처리 속도를 향상시킨다.

## 1. 서론

지리정보시스템에서 다루는 데이터의 양은 매우 방대하여 공간 질의를 처리하기 위해 많은 수의 디스크 참조를 요구하게 되고 이것은 질의 응답시간을 지연시키는 요인이 된다. 질의 처리시 데이터 탐색을 효율적으로 하기 위해 지리정보시스템에서는 공간 색인 방법을 도입한다[4].

본 논문에서 제안하는 공간 질의 향상을 위한 버퍼관리 기법은 공간 질의가 가지는 특성에서 출발한다. 공간 질의는 특정 시간에 특정 지역에 집중적으로 나타난다. 일반적으로 지리정보시스템에서 다루는 공간 객체들은 상호간에 공간적 근접성을 띠고 분포하며 특정 지역에 집중되어 나타난다. 마찬가지로 공간 질의 또한 공간 객체가 집중된 특정 지역에 주로 나타날 것이고 이는 질의가 공간적 집중성을 보이게 됨을 말한다. 또한 공간 질의는 특정 시간에 집중되어 나타나기도 하는데 이는 공간 질의가 이전의 질의 결과와 연관성있게 나타나기 때문이다. 그러므로 이전의 질의 결과로 검색된 객체들은 또한 다음 질의에서도 검색될 가능성이 높다. 본 논문은 이러한 공간 질의의 시간적 공간적 집중성을 기반으로 질의 처리를 향상시키는 방법에 대해 연구한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 데이터베이스 시스템의 버퍼링방법에 대해 살펴보고 3장에서는 공간 질의의 시간적 공간적 집중성과 이러한 집중성을 이용한 영역 버퍼링에 대해서 다룰 것이고, 영역 버퍼링을 개선한 확장된 영역 버퍼링 또한 설명될 것이다. 기존의 블럭 버퍼링 방법과 본 논문에서 제안하는 영역 버퍼링의 성능은 4장에서 비교될 것이며 끝으로 5장에서는 본 논문의 결론을 맺고 앞으로의 연구 방향을 제시하도록 하겠다.

## 2. 기존의 블럭 지향적 버퍼링

디스크 참조 횟수가 중요한 성능평가의 요인이 되는 시스템들은 효율적인 버퍼관리를 요구한다. 즉, 앞으로 자신이 참조하게 될 가능성이 있는 디스크 블럭을 메인 메모리에 유지함으로써 디스크 참조를 줄이도록 할 수 있다. 효율적인 버퍼 관리는 그 시스템의 특성을 최대한 고려해야

한다. 데이터베이스 시스템에서는 앞으로 사용될 블럭이 가장 오래 전에 사용된 블럭이라는 것을 예측할 수 있는 경우가 있다. 이러한 경우는 종종 많은 처리시간을 요하는 조인등에서 발견할 수 있는데, 이 때는 기존의 운영체제가 제공하는 LRU보다 MRU페이지 교체 전략을 이용하는 것이 성능을 더 향상시킨다. 현재, 지리 정보 시스템을 위한 버퍼 관리에 대한 연구는 거의 이루어지지 않았고 단지, 기존의 데이터베이스 시스템과 같이 LRU기법을 주로 사용한다 [1]. 본 논문에서 영역 버퍼링과 비교될 기존의 버퍼링 기법은 LRU교체 기법을 기본으로 하는데, 이는 [7]에서 소개된 Path Buffering와 유사하다. 본 논문에서는 이 후 이 방법을 블럭 지향 버퍼링이라 칭한다. 그러나, 이 방법은 지리정보시스템의 특성을 전혀 고려하지 않고, 기존의 방법을 그대로 사용한 것에 불과하다. 위의 서론에서 언급되었듯이, 지리정보 시스템은 데이터들뿐만 아니라, 주어지는 질의들도 집중성을 가지고 있다. 이 특성을 이용한다면, 더 효율적인 버퍼관리를 하여 질의 처리의 성능을 향상시킬 수 있다.

## 3. 질의 영역 버퍼링

### 3.1 공간 질의의 집중성

공간 객체의 분포는 집중성을 띠게 된다. 즉 공간 객체는 주어진 공간에서 균일하게 분포되어 있기보다는 어느 특정 지역에 집중되어 있음을 의미한다. 한 도시의 지도를 보더라도 도시의 중심부에는 건물, 도로등의 데이터가 집중되어 있는 반면 도시 외곽으로 갈수록 이러한 데이터의 밀도는 작아지게 된다.

또한 공간 질의에도 마찬가지로 적용하면 공간 질의도 특정 지역이나 특정 시간에 집중적으로 발생함을 알 수 있다. 공간 질의는 질의가 주어지는 객체에 의해 영향을 받게 되는데 위의 도시 데이터 같은 경우 정보가 집중된 도시 중심부가 도시 외곽에 비해 높은 질의 가능성을 내포하고 있다고 볼 수 있다. 그리고 질의의 결과는 다음의 질의에도 영향을 주게 되는데 예를 들어, "특정 도시에서 백화점들 모두 찾아라." 라는 질의가 주어졌을 때 다음으로 받

생할 수 있는 질의는 주로 “백화점 근처에 극장이 있는 곳은 어느 백화점인가?” 또는 “백화점중 제일 가까운 곳은 어디인가?” 등이라고 예상할 수 있을 것이다. 이러한 경우는 일련의 연속적인 질의들이 검색하는 지역이 서로 근접해 있거나 일치하고 있으므로 질의의 시간적인 집중성과 함께 공간적인 집중성 [2]도 반영하는 한 예가 될 수 있다.

공간 질의는 질의의 대상이 되는 공간 객체의 분포와 관련하여 주어진다. 공간 객체의 밀도가 높은 지역은 검색할 객체가 다양하고 정보량도 많아서 자주 검색지역에 들 가능성이 높은 반면 밀도가 낮은 지역은 실제로 검색의 범위에 들 가능성이 낮다. 그러므로 전체적으로 볼 때 질의는 공간 객체의 밀도가 높은 지역에 공간적으로 집중되어 나타나게 된다. 이 사실은 다음의 가정을 유도한다.

**Assumption 1 (질의의 공간적 집중성)** 공간 객체가 차지하는 영역  $A_0, A_1, \dots, A_N$ 과 질의 영역  $Q_0, Q_1, \dots, Q_N$ , 이 주어져 있고 공간 객체의 밀도가 다음과 같을 때

$$Density_o(A_0) < Density_o(A_1) < \dots < Density_o(A_N)$$

다음의 관계가 성립한다.

$$Density_q(Q_0) < Density_q(Q_1) < \dots < Density_q(Q_N)$$

그러나 [가정 1]은 공간 질의의 공간적 집중성은 나타내지만 시간적 집중성에 대해서는 전혀 설명하고 있지 않다. 공간 질의는 시간적으로 이전의 질의와 그 결과에 관련하여 주어진다. 왜냐하면 일련의 질의들은 서로 아무런 연관 없이 들어오는 것이 아니라 데이터베이스시스템에서 얻고자 하는 정보같은 특정 목적을 위해 주어지게 된다. 그러므로 이전의 질의에 의해 검색된 공간이 다음 질의에 다시 검색될 가능성이 높고 이전 질의의 결과로 검색된 객체들 중에서 다음 질의를 만족하는 객체가 존재할 가능성 또한 높다. 이는 공간 질의가 시간적 순서에 의한 집중성을 가진다는 다음의 [가정 2]를 도출하게 된다.

**Assumption 2 (질의의 시공간적 집중성)**

공간 질의  $Q_0, Q_1, \dots, Q_N$ 이 주어져 있고 질의 사이에 다음의 관계를 만족할 때

$$Distance(Q_i, Q_j) < Distance(Q_i, Q_k)$$

다음의 관계가 성립한다.

$$Prob(Q_i = Q_j) \geq Prob(Q_i, Q_k)$$

즉, 질의가 시간적 집중성을 띠는 것은 질의로 주어지는 영역이 서로 가깝게 위치할 경우가 멀리 떨어진 경우보다 연속적으로 발생할 확률이 높다는 것이다.

이와 같이 일련의 질의들은 공간 객체의 분포와 질의의 시간적 순서에 의해 공간적 시간적 집중성을 가지게 되고 본 논문에서는 이를 근거로 하여 질의 처리시 디스크 참조를 줄이면서 효율적으로 질의를 수행하는 기법을 개발한다. 본 논문에서는 이를 질의 영역 버퍼링이라 칭한다.

### 3.2 공간 질의의 집중성을 이용한 질의 영역 버퍼링

영역 버퍼링은 공간 질의의 집중성을 이용하기 위해 이전의 질의에서 검색되는 모든 공간 객체들을 버퍼에 저장하여 다음의 질의에서 이를 참조할 수 있도록 한다. 본 논문에서 다루는 질의의 종류는 영역 질의에 국한되어 특정 영역과 겹치는 공간 객체들을 검색하도록 한다.

영역 버퍼링은 공간 질의의 집중성을 이용하기 위해 이전의 질의에서 검색되는 모든 공간 객체들을 버퍼에 저장하여 다음의 질의에서 이를 참조할 수 있도록 한다. 그림 1에서 보면 질의로 들어온 영역 QR1과 QR2가 버퍼에 저장되어 있다. 버퍼는 질의 영역과 페이지 교체 전략을 위한 참조 계수 그리고 질의를 만족하는 객체들로 구성된다. 공간 객체들의 수는 일정하지 않으므로 필요할 때마다 버퍼 크기 한도 내에서 할당받는다. QR2는 QR1과 겹치는 부분을 제외 한 나머지 영역의 좌표와 그 안에 존재하는 객체들을 버퍼에 저장한다. QR3가 다음 질의로 수행될 때 먼저 영역 버퍼에서 QR1과 겹치는지 비교하여 겹치지 않는

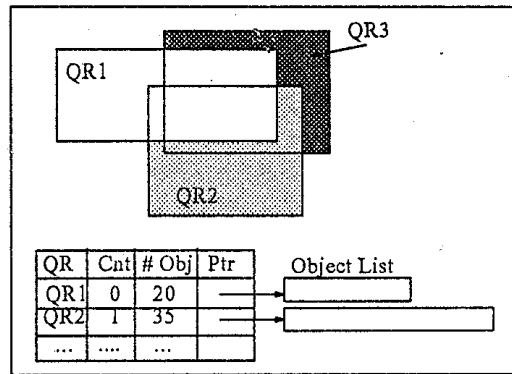


그림 1: 질의 영역 버퍼링 기법

나머지 영역을 구한다. 다음 QR2와 비교하여 역시 QR2와 겹치지 않는 부분을 구하게 된다. 이 경우 남은 질의 영역 즉 빗금친 영역은 공간색인을 참조해야 하고 QR3의 부분 중 버퍼와 겹치는 부분은 메모리내에서 검색하게 된다. 공간 질의의 시공간적 집중성에 의해 질의 영역들은 충분히 겹칠 것이고 그러므로 디스크 참조 횟수는 줄어들 것이다.

새로운 영역을 버퍼에 저장해야 할 경우 더 이상 버퍼에 공간이 없을 때 버퍼에 저장된 영역과 교체를 해야 한다. 이전에 저장된 영역들 중 어느 영역을 교체할 것인가는 다음에 그 영역이 참조될 가능성을 고려해야 한다. 공간 질의의 시간적 공간적 집중성을 생각하면 공간 질의가 특정 시간에 특정 지역에 집중되어 나타나므로 가장 최근에 참조되지 않은 영역을 교체 영역으로 선택하는 것이 타당한 방법일 것이다. 이는 기존의 운영체제나 데이터베이스 시스템에서 사용되는 LRU기법을 의미한다. 또한 공간 객체를 저장하는데 사용되었던 메모리가 반환되었을 때 새로 저장될 공간 객체들을 위한 메모리를 할당받을 수 있을 만큼의 공간을 확보할 수 있어야 한다. 이것은 질의 선택되는 공간 객체의 수가 일정하지 않기 때문이다. 이러한 메모리 문제를 LRU기법을 약간 변형한 방식으로 구현한다. 영역 교체를 가장 최근에 참조되지 않은 영역을 우선으로 하고 새로운 공간 객체들을 저장할 수 있는 메모리 공간이 확보되면 교체를 한다. 메모리 공간이 확보되지 않으면 다음 순위의 참조 계수를 가진 영역을 찾고, 경우에 따라서는 두 개 이상의 영역을 버퍼에서 교체해야 새로운 영역을 저장할 수 있다. 또한 필요한 메모리 크기가 버퍼의 크기를 능가할 때에는 버퍼에 유지할 수 없으므로 객체들을 저장하지 않는다.

질의 처리에 영역 버퍼링 기법을 사용하는 알고리즘은 알고리즘 1과 같다.

#### Algorithm 1 (영역버퍼링 알고리즘)

RegionBuffering(RB:RegionBuffer, QR:QueryRegion)

```

if QR ⊆ RB then
    processQuery(QR) in RB;
    return;
else if QR ⊇ RB then
    delete RB contained by QR;
    processQuery(QR) with Index;
    AddToRB(QR);
else if QR ∩ RB ≠ ∅ then
    processQuery(QR) in RB;
    processQuery(QR - RB) with Index;
    AddToRB(QR);
else // QR ∩ RB = ∅ //
    processQuery(QR);
    AddToRB(QR);
End RegionBuffering
    
```

3.3 확장된 질의 영역 버퍼링

질의 영역 버퍼링 기법은 공간 질의가 이전의 질의 영역내에 존재하게 될 때 디스크 참조 없이 질의 처리가 가능하게 되어 좋은 효율을 보이게 된다. 그러나 그림 2에서 보듯이 질의들이 시공간적 집중성이 있어도 질의 영역이 작으면 서로 겹치는 부분 또한 작아져서 다음 질의들이 히트할 확률이 작다. QR1을 버퍼에 유지하더라도 연속적인 질의 QR2, QR3 등의 질의들이 디스크 참조를 계속 요구하게 된다. 이는 버퍼에 존재하는 영역과 다음의 질의가 잘 겹치지 않아 버퍼를 효율적으로 사용하지 못한다는 것이다. 공간 질의는 시공간적 집중성에 의해 특정 지역에 집중적으로 발생하므로 이전 질의가 주어졌던 지역 근방에 다음의 질의들이 주어질 것임을 예상할 수 있다. 그러므로 질의 영역의 크기가 작을 때 질의 영역을 확장하는 방법을 사용하여 다음의 질의들을 포함할 수 있도록 하면 버퍼를 효율적으로 사용할 수 있게 된다. QR1을 원래의 영역보다 점선의 영역까지 확장하여 처리하면 질의 처리에 시간을 좀 더 요구하게 되나 다음의 QR2, QR3 등의 질의들은 버퍼를 통한 처리가 가능하다. 이렇듯 확장된 질의와 그 속에 존재하는 공간 객체들을 미리 버퍼에 유지함으로써 다음의 질의가 히트될 수 있는 확률을 높이고 디스크 참조를 하지 않고 버퍼에서 공간 객체들을 검색하도록 한다. 확장된 질의 영역 버퍼링 기법에서의 버퍼 크기에 따른 확장 비율은 다음 장에서 측정될 것이다.

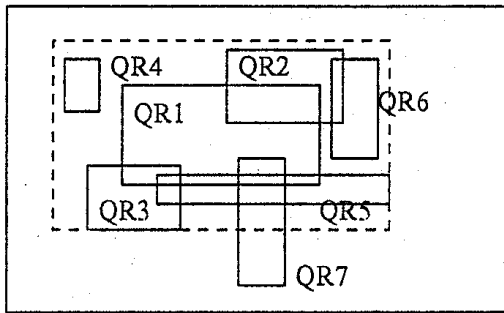


그림 2: 확장된 질의 영역 버퍼링 기법

4. 성능 측정

이 장에서는 2장의 기존의 방법에서 소개된 블럭 지향 버퍼링 (이하 비교표와 그래프에서 BOB(Block Oriented Buffering)라 칭한다.)과 본 논문에서 제시된 질의 영역 버퍼링(이하 비교표와 그래프에서 QOB(Query region Oriented Buffering)라 칭한다.)의 성능을 비교, 측정하였다. 버퍼를 효율적으로 사용하여 디스크 참조를 줄이는 것은 결국 주어질 질의 처리 시 메모리 히트율을 높이는 것이 된다. 따라서 본 논문에서는 모든 실험에서 다음과 같이 정의된 히트율을 조사하여, 성능 평가의 기준으로 삼는다.

$$hit\ ratio = \frac{N(Q_{NA})}{N(Q)}$$

$N(Q_{NA})$ : 디스크 참조를 하지 않는 질의의 수  
 $N(Q)$ : 총 질의의 수

각 실험에 쓰이는 공간 객체는 다각형 객체로써 10000개이며, 각 객체의 면적은 전체 면적의 0.0001배이다. 이는 균등하게 분포된 것도 있고, 특정 지역에 집중된 것도 있다. 그리고 공간 색인기법으로는 R\*-tree [6]를 사용하였다.

먼저, 각 질의의 크기와 버퍼로써 사용될 수 있는 메모리의 크기를 변화시켰을 때, QOB와 BOB의 히트율을 조사하였다. 데이터는 균등하게 분포 되고, 어느 정도의 집중성을 가지는 500개의 질의를 사용하였다. 표에서 사용된 변수인 버퍼크기비율은 다음과 같다.

$$buffer\ size\ rate = \frac{B}{S}$$

B: 버퍼의 크기

S: 전체 데이터의 크기

이 실험에서 사용된 객체들의 총 크기는 3M 바이트이므로, 버퍼 크기는 순서대로 8M, 16M, 32M, 64M, 128M 바이트이다. 그리고, 질의 처리 시 결과가 되는 객체의 갯수가 평균 6.4개인 500개의 질의들을 small이라 하고, 31.1개인 것을 middle, 66.9개인 것을 large로 정하였다.

buffer size rate	query size					
	small		middle		large	
	QOB	BOB	QOB	BOB	QOB	BOB
0.002	0.005	0.0	0.003	0.0	0.0	0.0
0.005	0.007	0.012	0.008	0.0	0.003	0.0
0.010	0.03	0.04	0.02	0.007	0.008	0.002
0.021	0.07	0.1	0.07	0.04	0.03	0.012
0.042	0.17	0.22	0.17	0.10	0.08	0.047
0.085	0.313	0.57	0.403	0.36	0.308	0.22

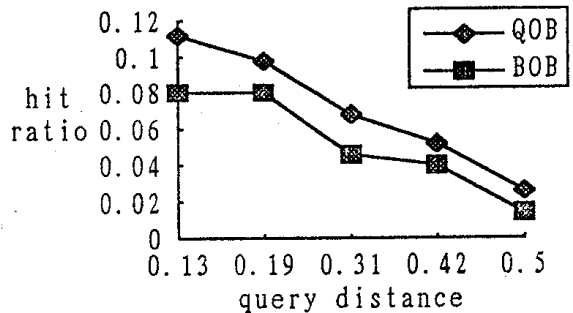
표1: 버퍼 크기와 질의 크기 변화에 따른 히트율의 비교

위의 표에서 질의의 크기가 small일 경우를 제외하고는 버퍼의 크기가 변하더라도 거의 대부분 QOB가 BOB보다 좋은 결과를 보인다. 이는 질의의 크기가 작으면, 질의 영역들이 서로 교차될 가능성이 작아지기 때문이다.

다음에 나오는 그래프1은 주어진 10000개의 객체들이 균등하게 분포되었을 때, 500개 질의들의 거리를 변화시켜서 얻은 히트율을 나타낸 것이다. x축에 사용된 질의의 거리인 query distance는, 질의 영역이  $Q_0, Q_1, Q_2, \dots, Q_{n-1}$ 와 같이 주어졌을 때, 다음과 같이 정의된다.

$$query\ distance = \frac{\sum_{i=1}^{n-1} (Q_i - Q_{i-1})}{n}$$

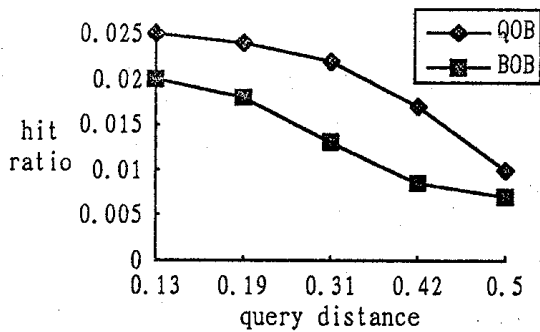
즉, 질의들간의 시간적, 공간적 집중성이 커질 때 그 거리가 작아지게 됨을 알 수 있다. 입력되는 질의의 순서를 조합하여 바꿈으로써, 서로 다른 거리를 가지는 일련의 질의의 집합들을 얻을 수 있다. 본 논문에서는 전체 영역의 크기를 1로 정규화 시켰다. 따라서, 질의 거리가 0.5에 가까운 질의의 집합은 임의로 질의를 생성한 경우를 의미한다.



그래프 1: 균등 분포를 가진 데이터의 질의 거리에 따른 히트율 비교

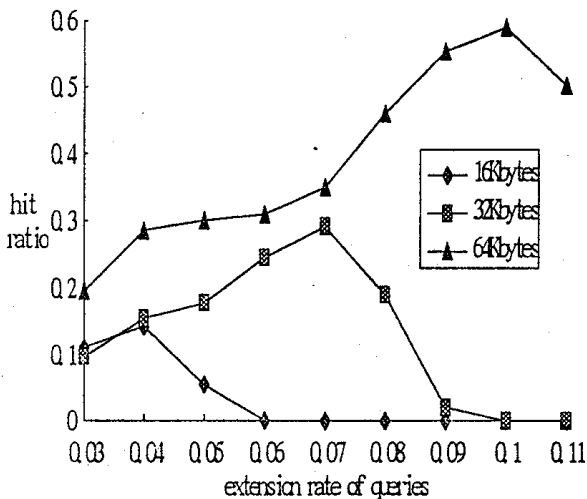
버퍼의 크기는 128K 바이트이며, 질의의 크기는 middle 이다. 그래프 1에서 우리가 예측했던 대로 QOB와 BOB 둘 다 질의의 거리가 작을수록 좋은 성능을 나타낼 수 있다. 또한, QOB가 BOB보다 더 높은 히트율을 나타내며, 집중성이 클수록 즉, 질의의 거리가 작을수록 차이가 더 커진다.

그래프 2는 사용된 공간 객체들이 실제계와 유사하게 어느 정도 집중성을 가진다는 점을 제외하면, 그래프 1과 같은 실험 환경을 가진다. 여기에서도 우리는 그래프 1과 비슷한 결론을 얻을 수 있다.



그래프 2: 비균등 분포를 가진 데이터의 질의 거리에 따른 히트율 비교

앞장에서 질의 영역 버퍼링을 개선한 확장 질의 영역 버퍼링에 대해 살펴 보았다. 성능 개선을 위해 주어진 질의들을 확장할 수 있다. 확장을 많이 할수록 좋지만, 확장된 질의의 결과들을 저장할 만한 버퍼가 없으면, 오히려 확장하지 않았을 때보다 성능이 훨씬 저하된다. 따라서, 주어진 버퍼의 크기와 수행될 질의의 평균 결과 데이터 수를 잘 고려하여 확장해야 한다. 그래프 3은 질의 영역 버퍼링에서 별로 성능이 좋지 못했던 small 사이즈의 질의에 대해서 버퍼의 크기를 변화시켰을 때의 성능의 변화들 보여 준다.



그래프 3: 확장 질의의 영역 버퍼링에서 확장을 변화에 따른 히트율

전체 영역의 가로,세로를 1로 정규화 하였을 때 확장을 에 따라 확장된 사각형 질의의 면적은 다음과 같다.

$$S_{ext} = (S_{org}.W + 2 * extensionratio) + (S_{org}.H + 2 * extensionratio)$$

$S_{ext}$  : 확장된 질의의 면적  
 $S_{org}.W$  : 원래 질의의 가로

$S_{org}.H$  : 원래 질의의 세로

버퍼가 16K 바이트일 때는 확장을 값이 0.04부근일 때 가장 좋은 성능을 나타내고 그 이후는 계속 감소하다가 어느 지점에서는 히트율이 계속 0임을 알 수 있다. 이는 본 논문에서 질의의 결과 버퍼 크기보다 많은 객체를 가지고 있을 경우 버퍼링을 하지 않기 때문이다. 즉 계속 버퍼를 증가시키는 것은 현실적이지 못하다. 따라서, 적당한 확장을 찾아야 한다. 그래프 3에서도 알 수 있듯이 버퍼의 크기가 커질수록 버퍼에 유지할 수 있는 결과 데이터의 갯수가 커지므로, 히트율이 가장 좋아지는 확장을 값 역시 커진다는 것을 알 수 있다.

5. 결론 및 향후 연구 방향

본 논문에서는 지리정보시스템에서 주어지는 질의들의 시간적, 공간적 집중성을 이용하여 질의 처리의 성능을 향상시키는 방법을 제안하였다. 이전의 질의의 결과가 되는 객체들을 버퍼에 저장하여, 다음의 질의 처리시 만약, 그 전의 질의들과 시간적, 공간적 연관성이 있을 경우 디스크를 참조하지 않고 버퍼에 가지고 있는 객체들을 조사하는 방법이다. 또한, 순서를 가진 질의들이 완전히 겹치지 않고 약간씩만 겹칠 때 생기는 연산의 과부하를 줄이기 위해 확장 질의의 영역 버퍼링 방법을 제안하였다. 제안된 두가지 영역 버퍼링 방법은 기존에 사용되는 가장 간단한 방법인 블럭 지향 버퍼링과 비교, 측정되었다. 질의사이의 거리를 정의하고, 버퍼의 크기와 질의의 크기 및 거리, 그리고 확장 질의의 영역 버퍼링에서의 확장을 변화시켰을 때의 성능을 측정하였다.

앞으로의 연구 과제는 성능 측정에 쓰인 여러가지 변수들에 대한 히트율의 분석적인 모델을 세워서 주어지는 질의에 대한 히트율을 예측할 수 있도록 하는 것이다. 또한, 버퍼 교체시 본 논문에서 사용된 LRU 교체 기법과 다른 기법들의 성능을 비교, 분석하는 것도 고려해 볼만하다.

참고 문헌

- [1] Henry F.Korth and Abraham Silberschatz DATABASE SYSTEM CONCEPTS. McGraw-Hill, 1991
- [2] H. M. Deitel An Introduction to Operating Systems. Addison-Wesley, 1983.
- [3] N. Roussopoulos, N. Economou, and A. Stamenas ADMS: A Testbed for Incremental Access Methods. IEEE Trans. on Knowledge and Data Engineering, vol.5, No.5, pp. 762- 73,1993
- [4] R.Laurini, D. Thompson Fundamentals of Spatial Information Systems. Academic Press, 1992
- [5] O. Guenther and A. Buchmann Research Issues in Spatial Databases. SIGMOD Record. Vol.19 No.4, 1990, pp.61-68
- [6] N. Brinkhoff, H.-P. Kriegel, R. Schneider, and B. Seeger The R\*-tree: An Efficient and Robust Access method for Points and Rectangles. Proc. SIGMOD'90, pp.322-331 1990
- [7] T. Brinkhoff, H.-P. Kriegel Efficient Processing of Spatial Joins Using R-tree. Proc. SIGMOD'93, pp.237-246 1993