고차원 공간에서 최근접 정의의 효율적으로 처리하기 위한 새로운 인덱싱 기법

김상욱¹  Charu Aggarwal¹  Philip S. Yu*  
¹ 강원대학교 컴퓨터정보통신공학부, ² Software Tools and Techniques Team IBM T.J. Watson Research Center  
wook@kangwon.ac.kr, (charu, psyul)@us.ibm.com

A New Indexing Technique for Processing Nearest Neighbor Queries in High Dimensional Space

Sang-Wook Kim²  Charu Aggarwal¹  Philip S. Yu*  
² Department of Computer, Information and Communications, Kangwon University  
Software Tools and Techniques Team, IBM T.J. Watson Research Center

요 약  
최근접 정의(nearest neighbor query)는 멀티미디어 데이터베이스에서 주어진 점의 객체와 가장 유사한 객체를 찾기 위하여 가장 널리 사용되는 질의의 하나이다[Ber98][Kor96][Roud95]. 기존의 최근접 정의 처리 기법들은 대부분 객체들의 효과적인 인덱성을 위하여 다차원 인덱스(multidimensional index)를 사용한다. 그러나 N차원 사각형 혹은 원을 사용하여 객체 클러스터의 범위를 표현하는 기존의 다차원 인덱스들은 차원 수가 높아짐에 따라 검색 성능이 크게 떨어진다. 본 논문에서는 이러한 문제를 해결하기 위한 새로운 인덱스 구조를 제시하고, 이를 이용하는 최근접 정의 처리 방안을 제안한다. 또한, 다양한 실험에 의한 성능 평가를 통하여 제안된 기법의 우수성을 검증한다.

1. 서론 ¹)  
최근접 정의(nearest neighbor query)는 멀티미디어 데이터베이스에서 유사한 객체를 찾기 위하여 가장 널리 사용되는 질의의 하나이다[Ber98][Kor96][Roud95]. 기존의 최근접 정의 처리 기법들은 대부분 객체들의 효과적인 인덱성을 위하여 사각형(Bec90), 원(Ber96) 등의 사각형 구조를 가지는 다차원 인덱스(multidimensional index)를 사용한다. 그러나 기존의 다차원 인덱스들은 차원 수가 높은 경우 좋은 성능을 나타내지 못하고, 차원 수가 높아짐에 따라 검색 성능이 크게 떨어지기로 알려져 있다[Web88][Ber96]. 본 논문에서는 객체 클러스터(cluster)의 범위(capsule)를 표현하는 새로운 방식을 제안함으로써 최근접 정의는 효과적으로 처리할 수 있는 새로운 인덱싱 방법을 제시하고자 한다. 또한, 사각형, 원의 검색 등 기존 기법들과의 성능을 통한 성능 평가에 의하여 제안된 기법의 우수성을 검증한다.

2. 동기  
최근접 정의는 처리하기 위한 기존의 기법들을 [Ber98][Kor96][Roud95]은 다차원 인덱스를 이용하여 전체 객체들을 다수의 객체 클러스터(cluster)들의 집합으로 구분하고, 주어진 정의의 집합으로부터 현재까지 구한 최근접 격자보다 있는 객체는 클러스터의 미리 작성 대상에서 제외하는 방식을 사용한다. 다차원 인덱스의 각 엔트리의 객체들을 포함하는 N차원 개별 옥(capsule)을 사용하여 클러스터를 표현한다. 좋은 검색 성능을 얻기 위해서는 객체의 클러스터의 간격을 모두 포함되며, 같은 영역(dead space)을 [Ber96]의 크기가 가능한 작게 명시되어야 한다.

3. 제안하는 기법
3.1. 축 시스템 변경  
본 논문에서는 각 열의 영역이 증가하는 문제를 해결하는 방법으로서 해당 클러스터에 맞는 새로운 축 시스템(=axis system)을 사용할 것을 제안한다. 그림 3.1은 그림 2.1에 나타난 클러스터를 새로운 축 시스템과 사각형 범위를 이용하여 표현한 것이다. 실제로 새로운 축 시스템을 사용하는 경우, 혹은 영역의 크기가 현저하게 감소함을 볼 수 있다. 차원의 수가 커질수록 이러한 측면의 감소 효과는 더욱 크게 작용된다.

1) 본 연구는 한국과학회계 99 해외 Post-Doc 방문 연극 프로그램의 지원을 받았습니다.

(a) 사각형 형태의 범위, (b) 원 형태의 범위, 그림 3.1. 관계가 있는 클러스터에서 범위의 형태와 측면의 관계.
3.2. 격렬한 모델의 특징

각 채널을 표현하기 위해서 원의 사각형을 조합함으로써 모양의 크기에 따라 특성화 방식을 제안한다. 먼저, \( r_1, r_2, \ldots, r_k \)을 구한다. 여기서, \( r_i \)는 변환된 채널의 각각의 i번째 채널의 평균값이다. 이후부터는 각 채널의 분포 형태에 맞도록 변환된 새로운 채널을 기준으로 분류를 진행한다. 또한, 점의 평균상 \( r_1 \leq r_2 \leq \cdots \leq r_k \)가 가정한다. 다음에는 \( r_i \)를 \( m(\sqrt{k})\)개의 그룹 \( r_1, \ldots, r_k \)으로 분할한다. 여기서, \( r_i \)는 \( r_{(i)} = \left( r_{(1)}, r_{(2)}, \ldots, r_{(n)} \right) \)로 분할한다. 이 경우, \( r_{(i)} \)는 \( r_{(1)}, r_{(2)}, \ldots, r_{(n)} \)의 평균을 구한 것이다. 예를 들어, \( r_{(1)} \)가 \( (1, 2, 3, 5, 8, 12, 15, 22, 26, 40) \)인 10차원 공간에서 클러스터의 경우, 위의 방법을 적용하면, \( (1, 2), (3, 5, 8), (12, 15), (22, 26, 40) \)의 세 그룹이 형성된다.

(a) 사각형 채널의 사용, (b) 사각형 채널의 사용.
그림 3.1. 변환된 채널의 사용.

클러스터를 위한 채널은 다음의 식들 표현하는 다차원 원들의 교집합으로 구성되는 공간에 의해 결정된다. \[ x_1 + \ldots + x_{(i)} = a_i, x_{(i+1)} + \ldots + x_{(k)} = a_k, \] \( x_{(k+1)} + \ldots + x_{(n)} = a_n \). 여기서, \( a_i \)는 \( i \)-번째 그룹에 의해 결정된 다차원 원의 지름을 의미한다. 그림 3.2는 3차원 공간에서 클러스터의 분포에 따라 가능한 모든 분류 방식과 이에 의한 채널의 형태를 도면화한 것이다.

그림 3.2. 3차원 공간에서 채널 형태의 결정.

3.3. 아웃라이어 처리

채널의 기본 형태는 분할의 결과에 의하여 결정되지만, 각 그룹의 형태를 결정하는 원의 지름 혹은 사각형의 변의 \( a_i \) 값을 의하여 결정한다. 가장 간단한 방법은 하나의 중심점과 이로부터 가장 멀리 떨어진 객체의 거리를 \( a_i \) 값으로 이용하는 것이다. 이 경우, 클러스터 내의 모든 객체들은 결정된 클러스터 내에 포함된다.

그림 3.3. 클러스터, 아웃라이어, 채널의 관계.

\[ a_i = R \times \left( \frac{N}{N+1} \right) \times \sqrt{N} \]

정리 1: \( k \)-차원 공간의 변가치가 \( R \) \( k \), \( N \) \( k \)-개의 객체들 사이의 긴급한 관계에서 \( N \) \( k \)-개의 객체들 사이의 거리를 결정한다. \( \mu \) 표준 편차 \( \sigma \)의 값을 다음과 같다.

\[ \mu = R \times \left( \frac{N}{N+1} \right) \times \sqrt{N} \]

\[ \sigma = R \times \left( \frac{N}{N+1} \right) \times \sqrt{N} \]

\[ \text{중평: 생략.} \]

3.4. 인덱스 구조 및 질의 처리 방안

그림 3.4에 나타난 바와 같이 제시된 인덱스 구조는 다수의 엔트리들로 구성되는 다이렉트리 패지지의 경계를 구성한다. 다이렉트리는 주기적으로 내부에 관리되며, 패지지와 다이렉트리 사이에서 관리된다. 각각의 다이렉트리 엔트리는 클러스터와 일체로 연결되며, 클러스터에 적합한 채널을 제공할 수 있다. 각 채널은 클러스터에 적합한 채널의 구조를 결정한다. 클러스터에 적합한 채널은 Principal Component Analysis에서 실질적으로 \( N \)-개의 기준을 사용하여 \( \Sigma \) (성분 분포의 고유값)의 크기 조정을 이용한다. 각 클러스터를 위한 샘플의 형태와 \( a_i \) 값을 결정한다. 그림 3.5는 클러스터에서 질의 범위에 존재하는 아웃라이어들을 탐색할 수 있는 인덱스 구조를 제시한다. 이 인덱스 구조는 값에 의한 클러스터의 관계를 결정하기 위해 다이렉트리 엔트리에 출력된다.
4. 성능 평가

제안된 기법의 비교 대상으로서 X-트리[Ber96]와 순차 검색 (sequential scan)을 채택한다. 메이저 크기로는 4k 바이트 를 세 가지 기법 모두에서 공통적으로 사용하였다. 제안된 기법 에서 X-트리의 엔트리의 수는 1,000으로 설정하였다. 실험에 서 사용된 객체 집합들은 2차원에서 100차원까지의 다양한 차 원 수로 가진다. 각 객체 집합은 플러스터의 집합을 기반으로 구성되며, 전체 100,000개의 객체들을 포함한다. 객체 집합은 각 플러스터 내의 객체 수에 따라 다음과 같이 분류한다.

- MC (many clusters): 500 ~ 1,000의 객체들을 가지는 많은 수의 플러스터들이 존재한다.
- FC (few clusters): 5,000 ~ 10,000의 객체들을 가지는 적 은 수의 플러스터들이 존재한다.

또한, 각 플러스터가 차지하는 공간내의 영역의 크기에 따라 다음과 같이 분류한다.

- LS (large standard deviation): 플러스터내의 객체들이 넓 은 공간 내에 분포한다.
- SS (small standard deviation): 플러스터내의 객체들이 좁 은 공간 내에서 분포한다.

![표 3.4. 인덱스 구조.](그림 3.4. 인덱스 구조.)


5. 결론

데이터베이스 테이터베이스 응용에서는 최근접 점의 효과적인 지정이 매우 중요하다. 이를 위하여 제안된 기존의 다차원 인 덕스들은 차원 수가 높아짐에 따라 성능이 급격히 저하되는 문제점을 가진다. 본 논문에서는 이러한 성능 저하가 플러스 터 클러스터의 쿼리 영역의 크기 증가로 인한 것임을 밝혀, 이 를 해결하기 위한 새로운 인덱싱 및 최근접 점의 처리 방안을 제시하였다. 제안된 기법의 우수성을 규명하기 위하여 다양한 실험을 통해 성능 평가를 수행하였다. 향후 연구 방향으로 는 (1) 제안된 인덱스를 트리 구조로 확장하는 방안, (2) 제안 된 기법과의 최근접 점의 처리 방안, (3) 최근접 점의 처리 점의 병렬 수행 방안, (4) 제안된 인덱스의 동적 환경으 로의 적용 방안 등을 고려하고 있다.

6. 참고문헌