클러스터링 정보를 이용한 R-tree 인덱싱 생성방안

라기용*, 김병곤*, 정헌석*, 이재호**,임해철*
*홍익대학교 컴퓨터공학과
**인천교육대학교 컴퓨터교육과
{kyra, bgkim, hschung, lim}@cs.hongik.ac.kr, jhlee@mail.inue.ac.kr

Mechanism of R-tree Indexing using Clustering Information

KiYong Ra*, ByungGon Kim*, HunSuk Chung*, Jaeho Lee**, HaeChull Lim*
*Dept. of Computer Engineering, Hong Ik University
**Dept. of Computer Education, Inchon National University of Education

요 약

최근 들어 멀티미디어와 같은 고차원 데이터를 효율적으로 처리하기 위한 고차원 인덱싱구조에 대한 연구가 활발히 진행되어왔으며, 특히 R-tree를 기반으로 하는 인덱싱 구조가 가장 많이 발표되었다. 그러나, R-tree 계열의 색인기법은 데이터 삽입시 삽입순서를 비효율적으로 배정하는 경우 실제적으로 자신과 거리가 먼 객체들과 동일한 노드에 삽입될 수 있다. 이는 인덱싱 구조내에 많은 겹침을 초래하고 결과적으로 검색성능을 저하시킬 수 있다. 본 논문은 이러한 단점을 극복하기 위하여 데이터가 지니는 특성 즉, 공간적인 위치정보를 분석하여 클러스터를 형성하고 이를 이용하여 인덱싱 형성에 적용하였고, 질의 처리시 검색성능이 향상되도록 하였다. 또한 실험에서는 실제 이미지 데이터에 대하여 제안한 기법을 적용하여 성능이 향상되었음을 나타내었다.

1. 서 론

인터넷의 확산과 웹 기술의 발전으로 이미지나 음성과 같은 멀티미디어 자료를 누구나 쉽게 이용할 수 있게 되었고, 다차원적인 특성을 가지는 멀티미디어 데이터를 다루기 위한 색인구조에 대한 연구의 필요성이 증가하고 있다. 이와 같은 다차원의 데이터를 효율적으로 처리하기 위하여 R-tree[1] 기반의 다양한 색인구조가 연구되었다. R-tree계열의 인덱싱 구조에서는 노드에 데이터 객체를 삽입할 경우 데이터의 삽입 순서에 따라 거리가 가까운 데이터들이 동일한 영역의 노드에 삽입되지 못하고 거리가 먼 다른 노드에 삽입되는 부적절한 할당이 발생할 수 있다. 이런 부적절한 할당은 노드간의 겹침을 증가시키고 질의 처리시 불필요한 노드의 검색을 증가시키므로 질의 성능을 저하시킨다. 그러나 어떤 특정한 정보를 가지고 데이터의 삽입순서를 조절하여 거리적으로 가까운 영역에 분포되어 있는 데이터를 동일한 노드에 삽입한다면 노드간의 겹침을 줄이고 트리 검색시 불필요한 노드의 검색비용을 줄일 수 있다. 삽입 순서를 결정하기 위한 특정 정보를 얻기 위해서는 데이터가 지니는 경향 및 특성 정보를 분석할 필요가 있다. 본 논문에서는 삽입 순서의 기준을 마련하기 위해 다차원 공간의 분포하고 있는 데이터들의 거리 정보를 사용하여 데이터들을 클러스터링[6,8] 하였고, 여기서 발생한 각 데이터의 소속 정보와 클러스터내의 중심값을 사용하여 데이터의 삽입 순서를 부여하였다.

본 논문의 구성은 다음과 같다. 2장에서는 R-tree 계열의 인덱스 구조와 적용한 클러스터링 방법에 대해서 간단히 살펴보고 3장에서는 클러스터링 정보를 트리구조에 적용하는 방법에 대해서 소개하고 4장에서는 동일한 데이터에 대하여 객체들의 입력순서에 따라 어떠한 결과를 보이는지 살펴보고 본 논문에서 제안한 클러스터 입력 방법의 우수함을 보였다. 비교 실험을 통한 성능 측정 결과를 보이도록 한다. 5장에서는 결론 및 향후 연구과제에 대해서 제시하였다.

본 연구는 한국과학재단 특정기초연구과제 (과제번호 : 98-0102-09-01-3)의 지원을 받았음

2. 관련연구

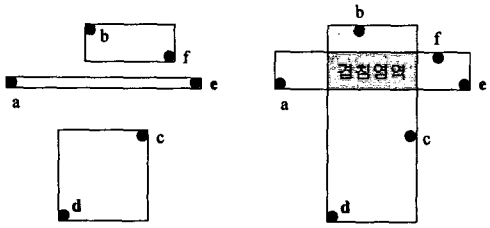
R-tree계열의 색인구조는 상위레벨의 최소경계영역(Minimum Bounding Rectangle)이 하위레벨의 최소경계영역을 포함하는 계층적인 구조를 지니며, 최소경계영역간에는 겹침(overlap)을 허용한다. 각 노드의 영역은 자식노드의 영역을 포함한 최소경계영역으로 결정되어지며, 단말노드의 영역은 데이터객체를 포함한 최소경계영역으로 결정되어 진다. 따라서 루트노드 영역은 모든 자료 엔트리들에 대한 최소경계영역이며, 중간노드의 영역은 그 자식들의 단말노드가 지니고 있는 데이터 엔트리들에 대한 최소경계영역이다.

R-tree[1] 색인 기법은 데이터 삽입 및 분열시 최소경계영역의 최소 증가만을 고려하기 때문에 데이터 삽입 순서를 잘못 배정한 경우 실제적으로 자신과 거리가 먼 객체들과 동일한 노드에 삽입 되어 단계를 거듭할 때마다 겹침 영역을 증가시키는 결과를 초래한다. 겹친 영역에 존재하는 데이터를 검색할 경우 데이터를 포함하는 모든 최소경계영역을 검색해야 되므로 질의 성능을 저하시킨다.

R+-tree[2]는 R-tree에서 발생하는 겹침을 최소화하기 위해 트리를 구성하는 중간노드에서 최소경계영역간의 겹침을 허용하지 않는 전략을

제시하였다. 그러나 너무나 많은 중간노드를 생성하게 되어 트리의 깊이가 증가하게 되며, 분할된 객체의 삽입과 삭제시에는 객체가 속한 모든 노드에 대하여 삽입, 삭제 작업을 해야 하므로, 트리 유지에 대한 오버헤드가 커지는 단점이 있다.

R*-tree[3]는 객체의 삽입과 삭제시 효율적인 공간의 재배치를 위한 강제재삽입을 수행한다. 또한 R-tree와는 달리 검색효율을 높이기 위한 변수로 영역, 겹침, 마진, 저장 장치의 효율성등을 고려하여 분할한다. 따라서 R-tree 보다는 데이터간의 위치 정보를 효율적으로 적용한 경우이다. 아래에 그림은 데이터의 삽입순서에 따라 서로 다른 겹침의 차이를 보여준다. a-b-c-d-e-f의 순서로 입력한 경우와 a-b-c-e-d-f의 순서로 입력한 경우의 겹침이 서로 다른 것을 알 수 있다. 본 논문은 이와 같이 겹침을 줄이는 객체 입력순서를 분석하기 위하여, 클러스터링을 이용하였다.



(a) a-b-c-d-e-f (b) a-b-c-e-d-f

[그림1] 입력 순서에 따른 겹침 차이

다차원 공간의 데이터를 클러스터링할 수 있는 클러스터링 알고리즘으로 유전자 알고리즘[6,8]이 있다. 유전자 알고리즘은 자연계의 진화 현상에 기초한 알고리즘으로 풀고자하는 문제에 적절히 적합도함수를 정의하여 이를 기준으로 적자만을 다음 세대로 생존시키므로써 세대가 거듭할수록 점점 더 좋은 해를 얻는 방법이다. 한 세대가 가지는 단계는 다음과 같다. 먼저 개체군을 초기화시키고 이 개체군을 가지고 선택(selection), 교배(crossover), 돌연변이연산(mutation) 및 평가(evaluation) 연산을 한다. 다시 선택(selection)과정부터 순서대로 적절한 해가 얻어질 때까지 같은 과정을 반복한다.

3. 클러스터링 정보를 사용한 인덱싱 생성

R-tree계열의 색인 구조에 삽입순서를 결정하기 위해 유전자 알고리즘을 사용하여 클러스터링[8] 하였다. 먼저 다차원 공간에서 임의의 개수의 데이터를 무작위로 추출하여 하나의 유전자 개체를 형성한다. 선택된 데이터는 각 클러스터의 중심값이 된다. 이와 같은 유전자 개체들이 모여서 개체군을 형성한다. 그 다음에는 유전자 개체를 평가하기 위해 사용되는 적합도 함수를 정의 한다. 이 함수를 기반으로 유전자 개체가 다음 세대에 적합한지를 결정한다. 적합도 함수는 다음과 같이 정의하였다. 개체군에서 유전자 개체를 구성하는 중심값들과 개체군내의 모든 데이터들과의 거리를 측정하여 데이터들의 소속 클러스터를 결정한다. 그리고 각 클러스터의 중심값과 그에 소속된 데이터들과의 거리의 합을 구한다. 이렇게 계산된 값들을 다시 합하여 그 결과값으로 개체를 평가하는 기준으로 사용한다. 개체 평가과정에서는 도출된 평가값을 기준으로 다음세대에 남을 개체를 선택한다. 개체 선택시 각 유전자의 적합도의 값의 크기에 비례하는 룰렛휠을 구성하여 무작위로 선택함으로 다양한 기회를 준다. 이렇게 선택된 유전자 개체들은 각자가 가지고 있는 중심값을 교배연산자를 사용하여 정의된 교배율의 범위 내에서 일부를 교환한다. 마지막으로 교배연산에서는 개체들의 사이에 유전 정보는 서로 교환 되지만 모든 해 공간을 탐색하기 위한 유전정보가 현재의 개체군내의 개체들에 존재하지 않는 다면 교배 연산을 아무리 적용시키더라도 더 이상의 탐색은 이루어 질 수 없다. 따라서 새로운 유전형질을 부여하는 돌연변이 연산자를 사용하였다. 이와 같은 과정을 반복하여 적합한 해를 찾을 때까지 계속 수행한다. 이렇게 얻어진 개체군내의 개체들을 적합도 값을 기준으로 정렬하여 가장 작은 값을 갖는 개체를 추출한다. 여기서 추출된 개체의 정보는 클러스터의 중심값과 데이터 객체들의 소속정보이며 이는 효과적인 R-tree 인덱싱을 위한 정보로 사용된다.

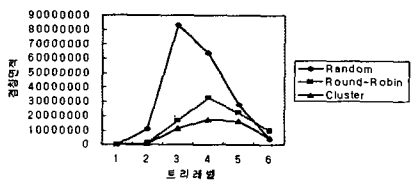
본 논문은 각 객체의 클러스터가 결정되면 각 객체들을 클러스터별로 분류하여 R-tree 인덱싱 생성시에 각 클러스터별로 입력하여 검색성능이 향상되도록 하였다. 4장에서는 제안한 클러스터 입력 방식이 인덱싱의 겹침 영역과 검색 성능면에서 향상되었음을 보여준다.

4. 실험 및 평가

R-tree 색인 구조에 데이터 삽입 순서를 결정하기 위해 클러스터링 과정에서 생성한 각 데이터들이 지니는 자신의 소속정보와 각 클러스터가 가지는 중심값을 이용한다. 이 정보들을 이용하여 세 가지 방법으로 삽입 순서에 변화를 준다. 첫 번째는 아무런 순서 없이 무작위로 데이터를 삽입하였다. 두 번째는 각 클러스터가 가지는 중심값을 기준으로 해당 클러스터 내에 있는 모든 데이터 객체들을 거리가 가까운 순서로 오름차순 정렬하였다. 그리고 정렬된 순서로 클러스터 단위로 모든 클러스터를 트리구조에 삽입하였다. 세 번째 방법은 노드엔트리 개수 만큼씩 라운드로빈 방식으로 클러스터를 돌면서 정렬된 순서를 유지하며 삽입하였다.

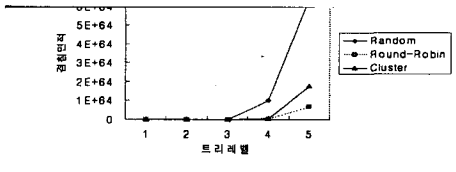
이와 같이 세 가지 방법으로 클러스터링 정보를 이용하여 구성된 R-tree 색인구조의 평가는 검색질의[4,7]시 노드의 방문회수와 관계가 있는 각 최소경계영역간의 겹침 정도를 기준으로 측정하였다. R-tree 계열의 트리구조는 하위노드가 상위노드에 포함되는 계층적 구조를 갖기 때문에 상위레벨의 겹침 정도는 하위 레벨의 겹침 정도보다 더 많은 검색비용을 유발한다. 본 실험에서는 클러스터정보를 사용하여 구성된 트리에 각각 레벨별로 해당 레벨에 있는 각 최소경계영역들간의 겹침을 중복없이 한번씩만 측정한 면적의 합을 측정 결과로 사용하였다. 실험에서 사용된 데이터는 30,000여개의 실제 이미지 데이터이다. 이미지 데이터에서 RGB 3가지 종류의 색상정보를 추출하여 3차원, 12차원, 27차원의 데이터를 생성하였다.

아래 실험은 30,000개의 이미지 데이터를 가지고 차원별로 3차원, 12차원, 27차원에 대해서 3가지 방법(무작위 입력, 라운드로빈 방식으로 입력, 클러스터 단위로 입력)으로 트리에 데이터를 삽입한 경우 트리의 겹침 영역을 계산한 결과를 나타낸 것이다.



(a) 3차원

(b) 12차원

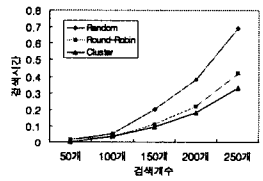


(c) 27차원

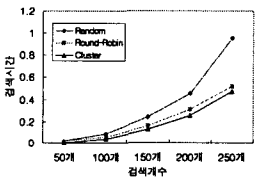
[그림2] 차원별 겹침 영역

위에 그림2에서 알 수 있듯이 데이터 입력시 클러스터링 정보를 이용한 방법 경우 즉, 클러스터별 입력과 라운드-로빈 입력 방법이 순서없이 무작위로 삽입한 경우보다 겹침 영역이 적음을 알 수 있다.

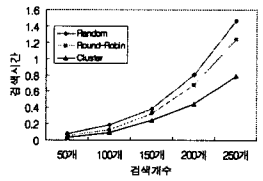
그림3은 동일한 데이터와 환경에서 임의의 질의에 대한 검색시간을 측정한 결과다. 질의 방법은 최근접객체질의(k-Nearest Neighbor)[7]를 사용하였고 주어진 질의 포인터에 대해 각각 50개, 100개, 150개, 200개, 250개의 데이터 객체를 검색하는데 걸리는 시간을 측정하였다.



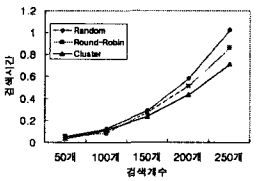
(a) 3차원, 질의1



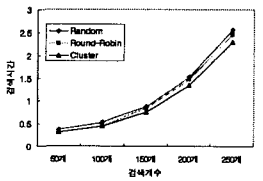
(b) 3차원, 질의2



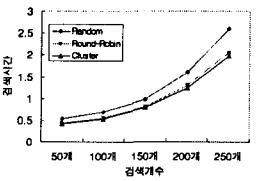
(c) 12차원, 질의1



(d) 12차원, 질의2



(e) 27차원, 질의1



(f) 27차원, 질의2

[그림3] 질의처리시간 측정

위에 그림3에서 알 수 있듯이 모든 차원의 데이터에 대하여 클러스링정보를 사용한 경우가 그렇지 못한 경우보다 적은 검색시간을 보임을 알 수 있다. 그리고 클러스터 정보를 사용한 결과들 중에도 클러스터 단위로 입력한 결과가 라운드로빈 방식으로 입력한 결과보다 적은 시간이 소요됨을 알 수 있다..

본 논문에서는 클러스터링 정보를 이용하여 R-tree 인덱싱 생성시 적용하였다. 무작위로 객체를 입력한 경우보다 일정한 규칙 즉, 클러스터링 정보를 사용하여 클러스터 단위로 삽입한 경우에 노드간 겹침이 감소하였고, 질의 처리시간 또한 단축되었음을 알 수 있었다. 향후 연구해야 할 부분은 다양한 적합도함수의 적용이다. 본 논문에서 사용한 클러스터링 적합도함수는 다차원 공간에서의 추상적인 거리만을 고려하였으나 데이터 공간의 거리뿐만 아니라, 객체의 면적이나 차원 등을 다양하게 고려하여 적합도함수를 구성하도록 할 수 있다.

5. 참고 문헌

[1] Antonin Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", Proceedings of the ACM SIGMOD Conference, pages 47-57, 1984.

[2] Timos K. Sellis, Nick Roussopoulos, Christos Faloutsos, "The R+-tree: A Dynamic Index for Multi-Dimensional Objects" , Proceedings of the VLDB Conference, pages 507-518, 1987

[3] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger, "The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles", Proceedings of the ACM SIGMOD Conference, pages 322-331, 1990.

[4] Nick Roussopoulos, Stephen Kelley, and Frederick Vincent, "Nearest Neighbor Queries", Proceedings of the ACM SIGMOD Conference, pages 71-79, 1995.

[5] Stefan Berchtold, Daniel A. Keim, and Hans Peter Kriegel, "The X-Tree: An Index Structure for High-Dimensional Data", Proceedings of the VLDB Conference, pages 28-39, 1996.

[6] Michael J. A. Berry Gordon Linoff, "Data Mining Techniques For Marketing, Sales, and Customer Support", Wiley Computer Publishing, pages 187-215 335-359, 1997.

[7] Thomas Seidl and Hans-Peter Kriegel, "Optimal Multi-Step k-Nearest Neighbor Search", Proceedings of the ACM SIGMOD Conference, pages 154-165, 1998.

[8] 유정우, 김명원,"진화알고리즘을 이용한 클러스터링 알고리즘", 한국정보과학회 제26회 가을 학술발표회, 1999