

효과적인 이미지 브라우징을 위한 M-트리 기반의 인덱싱 방법 (M-tree based Indexing Method for Effective Image Browsing)

유정수* 낭종호**
(Jeongsoo Yu) (Jongho Nang)

요약 본 논문에서는 대량의 사진 데이터베이스에 대한 효과적인 사진 검색을 위하여 브라우징을 지원하는 인덱싱 방법을 제안하였다. 제안한 방법은 대표적인 거리 공간 인덱싱 방법인 M-트리를 기본 구조로 하였다. 그러나 M-트리는 pruning을 통한 검색의 효율성에 초점을 맞추었으며 브라우징을 직접적으로 고려하지 않는다. 따라서 본 논문에서는 M-트리의 노드 선택 방법, 노드 분할 방법 및 노드 분할 조건을 브라우징에도 적합하도록 변경하였다. 제안한 방법을 적용하여 실험한 결과 노드 응집도와 클러스터링 정확도에서 각각 2배와 1.5배의 향상을 나타내었으며, 검색 성능도 약 2배정도 빨라지는 효과를 확인하였다.

키워드 : 인덱싱, 브라우징, 사진 검색, M-트리

Abstract In this paper we propose an indexing method supporting the browsing scheme for effective image search on large photo database. The proposed method is based on M-tree, a representative indexing scheme on matrix space. While M-tree focuses on the searching efficiency by pruning, it did not consider browsing efficiency directly. This paper proposes node selection method, node splitting method and node splitting conditions for browsing efficiency. According to test results, node cohesion and clustering precision improved

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음(NIPA-2010-(C1090-0903-0004))

· 이 논문은 2009 한국컴퓨터종합학술대회에서 '효과적인 이미지 브라우징을 위한 M-tree기반의 인덱싱 방법'의 제목으로 발표된 논문을 확장한 것임

* 학생회원 : 서강대학교 컴퓨터공학과
tool700@naver.com

** 종신회원 : 서강대학교 컴퓨터공학과 교수
jhngang@sogang.ac.kr

논문접수 : 2009년 8월 13일
심사완료 : 2010년 1월 11일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 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: 컴퓨팅의 실제 및 레터 제16권 제4호(2010.4)

1.5 and twice the original respectively and searching speed also increased twice the original speed.

Key words : Indexing, Browsing, Image Search, M-tree

1. 서론

최근 몇 년 동안, 디지털 카메라 및 카메라 폰의 보급이 급격히 증가함에 따라 대량의 디지털 사진에 대한 구조적인 관리를 통한 효과적인 사진 검색은 매우 중요하게 되었다. 그 동안 “query-by-search”는 대표적인 사진 검색 방법으로서, 사용자는 질의 데이터를 제공하여 이와 비슷한 사진에 대한 검색을 전체 데이터베이스에서 수행하게 된다. 또 다른 검색 방법으로는 브라우징 메커니즘을 들 수 있다. 브라우징은 사용자가 데이터 베이스에서의 네비게이션을 통해 원하는 사진을 구조적인 방식으로 찾게 된다[1-3]. 효과적인 네비게이션을 위해서는 연관된 다수의 사진에 대한 대표 사진을 통해 미리 보기를 제공해야 하며, 따라서 비슷한 사진들끼리 모으는 클러스터링을 필요로 한다. 예를 들어 전체 데이터를 트리 형태의 계층적 클러스터링을 통해 트리의 사용자는 트리의 각 레벨을 위아래로 오가며 TOC(table-of-contents) 효과를 얻을 수 있다. 이와 같은 “query-by-search” 또는 브라우징을 위해서는 데이터의 특징을 표현하는 공간과 각 특징 데이터간의 거리 측정이 필요하다. 영상 데이터를 기술하는 대부분의 특징 데이터들은 거리 공간에 존재하고 차원이 높으며 거리 측정의 계산 비용이 높다는 특징을 갖는다. 또한 새로운 데이터의 추가나 기존 데이터의 삭제로 인하여 데이터베이스가 지속적으로 변화하기도 한다.

본 논문에서는 대량의 사진 데이터베이스에 대한 효과적인 사진 검색을 위하여 브라우징을 지원하는 인덱싱 방법을 제안하였다. 제안한 방법은 대표적인 동적 거리 공간 인덱싱 방법인 M-트리[4]를 기본 구조로 하였다. M-트리는 동적 구축을 지원하고, 노드 크기를 고정함으로써 입출력의 최적화에 초점을 맞추었으며, bottom-up방식의 균형된 트리 구조를 통해 동적 삽입 및 삭제가 자유롭다는 특징을 갖는다. 그러나 M-트리는 기본적으로 pruning을 통한 검색을 목적으로 하고 있으며 브라우징을 직접적으로 고려하지는 않았다. 따라서 본 논문은 검색과 브라우징을 모두 지원하는 인덱싱을 위해서, M-트리의 구조를 기반으로 하면서 삽입 시 노드 선택 방법 및 노드 분할 조건, 노드 분할 방식을 변경 브라우징에 적합하도록 다음과 같이 변경하였다.

1. 삽입 시 노드 선택 - 리프 노드 중 노드에 포함된 오브젝트들과 삽입되는 오브젝트와의 평균 거리가 가장 가까운 노드.
2. 노드 분할 조건 - 노드가 오버플로우 이거나 노드 내의 오브젝트들간의 거리가 임계값보다 클 경우, 임계값은 트리의 레벨에 따라 정해진다.

3. 노드 분할 방식 - Slim Tree[5]의 분할 방식인 MST (minimum-spanning-tree)의 가장 긴 간선을 기준으로 분할.

이와 같은 기존 M-트리에 대한 변경을 통하여 향상된 브라우징 효과를 실험을 통하여 확인하였다. 실험 결과 노드의 응집도는 기존 M-트리의 0.012에서 0.027로 약 두 배 이상 증가하였으며 클러스터링 정확도는 0.45에서 0.7로 약 1.5배 증가한 것을 나타내었다. 뿐만 아니라 노드 응집도의 증가로 인하여 노드 간의 오버랩(overlap)이 감소함으로써 검색 성능도 약 2배정도 향상된 것으로 나타났다.

2. 관련 연구

멀티미디어 데이터에 대한 클러스터링의 경우 주로 사용되는 멀티미디어 데이터의 특징 데이터는 차원이 높고 거리 공간에서 정의되며 전체 데이터의 수가 지속적으로 변화한다는 점을 고려하여야 한다. R-트리[6]의 경우는 동적 변경을 고려하였지만 거리 공간에서는 좌표 시스템의 부재로 인하여 R-트리의 사각형 구역을 계산하기 힘들다. R-트리와 같은 공간 접근 방법(SAM, Spatial Access Method)기반의 인덱싱 방법들은 오브젝트의 좌표를 이용하므로 대부분 고차원 거리 공간에 존재하는 멀티미디어 데이터에 대한 인덱싱에는 적합하지 않다. 좀더 일반적인 유사도 인덱싱을 위하여 제안된 거리 접근 방법(MAM, Metric Access Method)기반의 인덱싱 기법들은 좌표 시스템을 요구하지 않으며 상대 거리만을 이용하므로 거리 공간에 존재하는 멀티미디어 데이터의 인덱싱에 적합하다는 특징을 갖는다. 그러나 기존의 MAM 방법들은 대부분 동적 삽입 및 삭제를 지원하지 않는다는 제한이 있다. 이러한 제한을 극복하기 위하여 bottom-up방식의 균형된 트리 구조를 사용하여 동적인 구축과 거리 계산 외에 입출력 비용을 고려한 M-트리기법이 제안되었다. M-트리는 좌표 공간에 제한하지 않으며 bottom-up 방식으로 트리가 성장하여 동적 삽입, 삭제가 가능하고 균형된 트리 구조와 고정 사이즈의 노드 크기 등 입출력의 최적화에 초점을 맞추었다는 장점을 갖는다.

M-트리는 좌표계를 정의하기 힘든 거리 공간을 지원하기 위하여 피벗(pivot)과 반지름(covering radius)을 사용한다. 또한 M-트리의 각 노드는 디스크 기반의 인덱싱을 고려하여 고정된 크기를 갖는다. 각 노드는 정해진 양(M 파라미터) 만큼의 엔트리를 포함한다. 노드는 내부 노드와 리프 노드의 두 가지 타입을 갖는다. 내부 노드의 엔트리(라우터 엔트리 or 포인터 엔트리)는 자식 노드의 포인터 역할을 하며 $\langle p, r^c, d(p, p^p), ptr \rangle$ 로 이루어진다. 여기서 p 는 피벗이고 r^c 는 p 가 가리키는 자식 노드의 엔트리 중 p 와의 거리가 가장 큰 값, 즉 자식

노드의 반지름을 나타내며, p 와 p^p 는 각각 부모 피벗과 그 거리를 나타내고, ptr 은 자식 노드에 대한 포인터이다. 리프 노드는 $\langle o, d(p, p^p) \rangle$ 형태의 엔트리를 포함한다. 이러한 M-트리에 대한 검색은 찾고자 하는 범위의 반지름이 r 일 경우 $|d(q, p^p) - d(p, p^p)| - r^c > r$ 인 피벗 p 가 가리키는 자식 노드는 방문하지 않음으로써 pruning 효과를 얻는다. 삽입 과정은 루트 노드에서부터 시작하여 가장 적합한 노드를 찾아가며 재귀적으로 내려가며 이루어지며, 리프 노드에 삽입된 엔트리가 노드의 반지름을 변경할 경우 상위 노드에 존재하는 포인터 엔트리의 r^c 값이 변경되고, 이와 같은 변경은 루트 노드까지 위 방향으로 발생한다. 또한 리프 노드에서 넘침 현상이 발생할 경우 노드는 분할하며 부모 노드에는 기존의 포인터 엔트리가 새로 분할된 두 노드에 대한 포인터 엔트리로 대체된다. 이때도 마찬가지로 다시 부모 노드가 넘침 현상이 발생할 경우 분할되며 루트 노드까지 위 방향으로 전달된다.

3. 브라우징을 위한 M-트리기반의 계층적 클러스터링 방법

본 장에서는 M-트리를 기본 구조로 하면서 M-트리의 노드 선택 방법, 노드 분할 조건 및 노드 분할 방법을 변경함으로써 검색 뿐만 아니라 브라우징에 적합하도록 계층적 클러스터링을 구축하는 방법에 대하여 설명한다.

A. 노드 선택 방법

M-트리에 새 객체를 삽입 시 루트 노드에서부터 리프까지 삽입하기에 가장 적합한 노드를 찾는 과정이 수행된다. 가장 적합한 노드란 M-트리의 pruning효과를 위하여 반지름과 노드 간의 겹침을 최소화 하는 노드이며, M-트리에서는 다음과 같은 정책을 제안하였다.

- 객체를 포함하는 노드가 1개일 때 : 해당 노드에 삽입
- 객체를 포함하는 노드가 2개 이상일 때 : 가장 가까운 라우터의 노드에 삽입
- 객체를 포함하는 노드가 존재하지 않을 때 : 새 객체를 삽입할 경우 가장 조금 확장되는 노드

그러나 본 연구의 목적인 가장 비슷한 객체들끼리의 클러스터링 기준으로 볼 때, 그림 1과 같이 위의 정책들이 적합하지 않은 경우들이 발생한다. 그림 1의 각각의 경우에서 새로 삽입되는 오브젝트 (검정색)은 오른쪽 노드의 오브젝트들과 비슷함에도 불구하고 M-트리의 삽입 정책에 따라 왼쪽 노드에 삽입되게 된다. 따라서 위와 같이 노드의 반지름이나 라우터의 위치 등을 기준으로 하는 방법 대신 (1)과 같이 노드에 속한 객체들과의 직접적인 비교를 통하여 삽입될 노드를 선택하는 방법을 사용하였다.

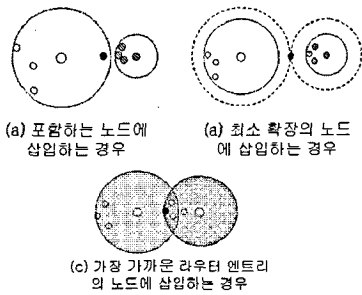
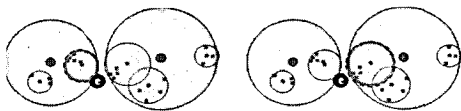


그림 1 브라우징에 적합하지 않은 M-트리 삽입 예, 새로 삽입되는 오브젝트(검정색)은 오른쪽 노드와 유사하지만 왼쪽 노드에 삽입된다.

$$\underset{\forall o_N \in N, \forall N \in \text{Leaf}}{\operatorname{argmin}} \operatorname{average}(d(o_N, e)) \quad (1)$$

(1)방법은 새로 삽입되는 오브젝트와 가장 가까운 오브젝트들을 포함하는 노드를 선택하는 방식으로, M-트리가 삽입 시 미리 계산된 값(반지름, 라우터로부터의 거리)을 사용하는 것에 비하여 노드 읽기 및 거리 계산 연산을 증가시키지만, 브라우징 시 사용자가 방문한 노드는 서로 비슷한 사진들을 출력함으로써 브라우징에 적합하다는 효과가 있다. 그림 2는 (1)을 사용할 경우 클러스터링 효과를 높일 수 있는 경우의 예시이다.



(a) M-트리의 삽입 (b) 제안한 방법의 삽입
그림 2 브라우징 효과가 향상된 노드 선택 예

또한 (1)방법을 사용할 때 모든 리프 노드에 대하여 계산하지 않도록 그림 3과 같이 M-트리의 기본적인 pruning 방법을 사용하여 비교 연산 횟수를 줄일 수 있다.

B. 노드 분할 조건

M-트리는 노드의 크기가 M 이상, 즉 오버플로우가 발생할 경우 분할을 실시하게 된다. 그러나 브라우징을 위해서는 비슷한 오브젝트들이 한 노드에 있어야 하므로, 노드가 오버플로우 상태가 아니라도 노드의 공간이 일정 수준 이상 커질 경우 서로 동떨어진 오브젝트가 한 노드에 존재하는 상태이므로 강제로 분할을 실시할 필요가 있다. 물론 이러한 방법은 노드의 분할 횟수를 증가시키고 노드의 수를 증가시키므로 검색 및 삽입 성능을 저하시킬 수 있는 가능성이 있다. 하지만 본 연구의 클러스터링은 다음과 같은 점들이 고려되어야 하기 때문에 노드 수를 증가시키더라도 이러한 방법을 사용하여 클러스터링 하였으며, 실제로 4절에서는 노드 수

```

// minDist is initially set with ∞
// ptr(o): the node pointed by o
// r(N): the radius of N
ChooseSubNode(N) {
  if N is Leaf {
    if average(d(o_N, e)) < minDist
      minDist ← average(d(o_N, e)), Nselected ← N
  } else {
    for ∀o_N | d(o_N, e) < minDist + r(ptr(o_N))
      ChooseSubNode(ptr(o_N))
  }
}
    
```

그림 3 브라우징에 적합하도록 제한한 노드 선택 방법, (minDist 보다 가까운 리프 노드만 비교)

가 증가하였지만 노드의 응집도가 높아졌기 때문에 검색 성능을 저하시키진 않은 것을 확인하였다.

1. 검색 성능 외에 브라우징 효과 증시.
2. 휴대 기기상의 사진 데이터는 일반 데이터 베이스에 비하여 전체 데이터 수가 훨씬 적으므로 검색 성능의 저하로 인한 역효과가 크지 않음.

노드의 응집 정도를 기준으로 분할 시점을 결정하기 위해서는 트리의 각 레벨에 따라 응집도의 기준을 계산할 필요가 있다. 왜냐하면 그림 4와 같이 단순 거리만을 가지고 응집도를 평가할 경우 각 레벨에 따라 노드간 거리의 스케일이 다르기 때문이다.

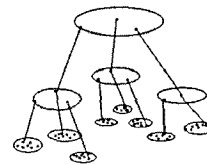


그림 4 레벨에 따른 노드 응집도

따라서 본 연구에서는 (2)와 같이 노드가 오버플로우이거나 또는 노드 내의 가장 먼 오브젝트간의 거리가 T_l 보다 큰 경우 분할하도록 하였다. T_l 은 Leaf인 경우 단순 거리 기준의 임계 값을 사용하고, Leaf가 아닌 경우에는 각 노드에서 동떨어진 엔트리들의 중간 값을 각 레벨에 포함된 노드들에 대하여 계산한 다음 이를 평균한 값을 분할 임계값으로 사용하였다. 이러한 방법은 각 레벨마다의 대략적인 노드간 거리의 스케일을 계산하기 위한 목적이다.

C. 노드 분할 방법

M-트리는 노드가 두 개로 분할 시 두 새로운 노드들의 라우터가 될 오브젝트를 선택하는 방법으로 mM_RAD 등의 몇 가지 promote policy를 제안하였다. mM_RAD의 경우 두 새로운 노드 중 반지름이 큰 노드의 반지름을

if $|N| \geq M$ OR $\max_{\forall o_i, o_j \in N} d(o_i, o_j) \geq T_i$

then split N

$$T_i = \begin{cases} \text{if } l=0 & T_d \\ \text{else} & \text{Average}(F_N | N \in S_l) \end{cases} \quad (2)$$

$$F_N = \text{Median}(d(o_r^N, o_i)), \forall o_i \in N | \sigma_N - \sigma_{N-o_i} \geq \theta$$

σ : standard deviation

S_l : set of node on level l

o_r^N : router node of node N

최소화하도록 두 엔트리를 선택한다. 또한 이와 같이 선택된 두 라우터 오브젝트에 대하여 나머지 오브젝트들을 가까운 라우터로 귀속시키는 unbalanced 분할 방법도 제안하였다. 이러한 M-트리의 분할 방식은 반지름을 최소화 하는 것이 목적이다. 그러나 그림 5의 예와 같이, 브라우징을 고려하면 라우터와의 거리뿐 아니라 인접 오브젝트간의 거리도 고려해야 한다. 따라서 그림 5(a)와 같은 클러스터링이 적합하다. Slim Tree에서는 M-트리의 새로운 분할 방식으로 MST(Minimum-spanning-tree)를 사용하는 방법을 제안하였다. 이 방법은 MST의 가장 긴 edge를 기준으로 분할함으로써 분할 속도를 향상시키고 각 노드를 더욱 컴팩트하게 유지할 수 있도록 하였다. 브라우징을 위해서는 노드의 반지름 보다는 노드 내의 오브젝트들의 응집도(오브젝트수/반지름)가 더 중요하므로 MST를 통한 분할이 더 적합하다고 볼 수 있다. 그림 5(c)는 이러한 예시이다.

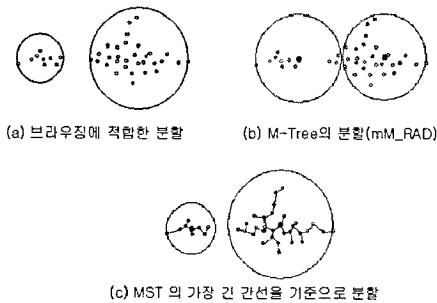


그림 5 분할 방법에 따른 브라우징 효과의 비교

4. 실험 결과 및 분석

본 장에서는 3장에서 제안한 방법들을 적용하여 브라우징 효과와 검색 성능을 측정할 실험 결과에 대하여 설명한다. 실험용 사진 데이터로는 www.cyworld.com, www.flickr.com 의 사진들로부터 각 500장으로 구성된 30개의 홈 포토 세트를 사용하였다. 또한 사진 간의 저수준 거리 측정을 위하여 [7]에서 사용된 블록 기반 영역 비교 방법을 사용하였다. 3장에서 제안한 방법이 브

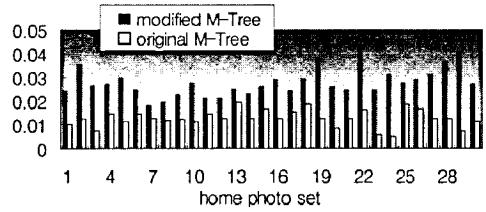


그림 6 노드 응집도 C_0 (30 홈 포토셋)

라우징에 적합한지를 평가하기 위하여 노드 응집도(3)과 클러스터링 정확도(5)를 측정하였다. 그림 6은 노드 응집도에 대한 결과이다. 기존 M-트리의 평균 응집도는 0.012이며 변경된 M-트리의 평균 응집도는 0.027로 2.25배 향상된 결과를 보였다. 각 홈 포토 세트는 3장 2절에서와 동일하다.

$$C_l = \text{Average}_{\forall N_l} \left(\frac{|N_l|}{N_l'} \right) \quad (3)$$

N_l : level이 l 인 노드, N_l' : 노드 N 의 반지름

또한, 클러스터링 정확도(4)를 비교하기 위한 실험을 아래와 같이 수행하였다. 클러스터링의 정확도를 측정할 때 특징 데이터에 대한 유사 거리의 오차가 적용되는 것을 방지하기 위하여, 사람이 직접 분류한 것 대신에 k-means clustering의 결과를 Ground Truth로 사용하였다. 그림 7은 제안한 방법들을 적용한 결과의 정확도가 기존의 M-트리 방법에 비하여 평균 약 1.5배 향상된 결과를 나타내었다.

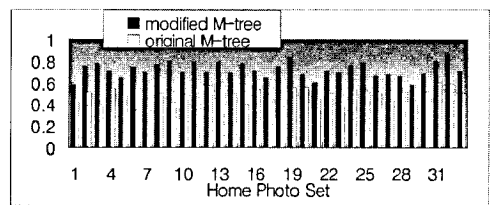


그림 7 클러스터링 정확도 $Accuracly_0$ (30홈 포토 셋)

$$Accuracly_i = \frac{1}{n} \sum_{\forall v_i} a_i$$

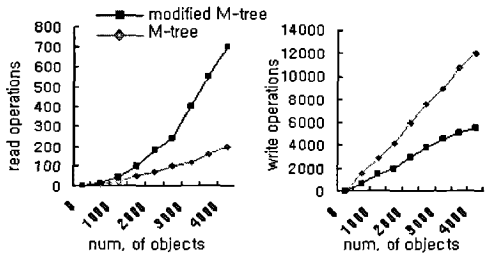
$$n: \text{the total number of objects, } a_i = \max_{\forall CL} \left| \frac{N_i \cap CL_i}{N_i} \right| \quad (4)$$

CL_j : ground truth

표 1은 총 5,000개의 오브젝트에 대한 인덱싱 삽입 비용 측정 결과로써, 제안된 방법을 적용하였을 때 삽입 시간이 약 30%정도 증가하였지만, 그림 3의 minDist를 통한 pruning 방법을 적용하였을 때 증가 폭이 약 10% 미만으로 완화된 것을 나타낸다. 그림 8은 표 1의 결과에 대하여, 입출력 비용과 거리 계산 비용에 대한 분석

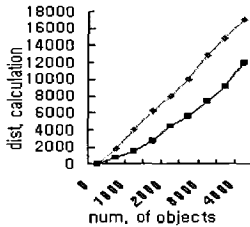
표 1 삽입 시간 결과(단위 : millisecond)

	평균 삽입 시간	총 삽입 시간
original M-tree	14.78	72,431
modified M-tree (without <i>minDist</i>)	19.47	97,619
modified M-tree (with <i>minDist</i>)	15.92	78,034



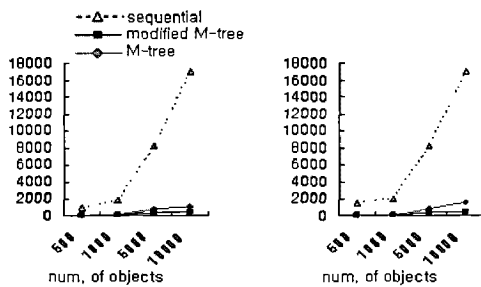
(a) 읽기 연산 횟수

(b) 쓰기 연산 횟수



(c) 거리 계산 횟수

그림 8 인덱싱 구축 비용 측정(단위:1,000)



(a) k-NN 검색 성능

(b) Range 검색 성능

그림 9 검색 성능 결과(단위 : millisecond)

이다. 그림 8의 (a)는, 3장의 모든 리프 노드와 비교하는 노드 선택 방법을 적용한 결과 읽기 연산 횟수가 크게 증가한 것을 나타낸다. 그러나 그림 8(c)에서, *minDist*를 적용하여 거리 계산 횟수가 감소된 것을 알 수 있으며, 또한 노드 응집도의 향상은 그림 8(b)에서와 같이 삽입 시 갱신 되는 노드의 수를 줄임으로써 쓰기 연산도 감소시키는 효과를 보였다. 그림 9는 검색 성능에 대한 결과이다. 15,000개의 사진에 대한 k-NN 검색과 Range 검색에서 각각 약 2배정도 성능이 향상된 것으로 나타

났다. 본 장에서 제안한 방법을 적용하였을 때 전체 노드 개수와 트리의 깊이는 기존 M-트리에 비하여 증가하게 되며, 이는 검색 성능을 저하시키는 원인이 될 수 있다. 하지만 각 노드의 응집도가 증가하였기 때문에 검색 시 *pruning* 효과가 높아지므로 검색 반응 속도가 크게 증가하였음을 알 수 있다.

5. 결론

본 논문에서는 M-트리 기반의 계층적 클러스터링을 구축 시 브라우징에 적합하도록, M-트리의 노드 선택 방법을 노드 내에 오브젝트들과 직접 비교하는 방법으로, 노드 분할 조건을 노드가 오버플로우이거나 노드의 공간이 임계치 이상으로 커졌을 때 분할하는 것으로 수정하였다. 또한 노드의 분할 방식은 기존의 Slim Tree에서 사용하는 MST를 이용하는 방식을 적용시킴으로써 좀 더 브라우징 목적에 적합한 분할이 가능하도록 하였다. 제안한 방법을 적용하여 실험한 결과 브라우징 효과와 검색에서 각각 성능 향상을 확인하였다. 향후 연구 과제로는, 노드 선택 시 계산량을 줄이며 좀 더 효율적인 구축이 가능하도록 하는 *pruning* 방법 등을 고려해 볼 수 있을 것이다.

참고 문헌

- [1] J. Chen, C. A. Bouman, and J. C. Dalton, "Hierarchical browsing and search of large image databases," *IEEE Transactions on Image Processing*, vol.9, no.3, pp.442-455, March 2000.
- [2] S. Krishnamachari and M. Abdel-Mottaleb, "Image browsing using hierarchical clustering," in *proceedings of the Fourth IEEE Symposium on Computers and Communications*, July 1999.
- [3] K. Barnard, P. Duygulu, and D. Forsyth, "Clustering art," in *Computer Vision and Pattern Recognition*, December 2001.
- [4] P. Ciaccia, M. Patella, F. Rabitti, and P. Zezula, "Indexing metric spaces with M-tree," *International Conference on Very Large Data Bases*, pp.426-435, August 1997.
- [5] C. Traina, Jr., A. J. M. Traina, B. Seeger, and C. Faloutsos, "Slim-trees: High performance metric trees minimizing overlap between nodes," in *proceedings 7th Int. Conf. EDBT 2000*, pp.27-31, March 2000.
- [6] A. Guttman, "R-Tree: A dynamic index structure for spatial searching," in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pp.47-57, June 1984.
- [7] James Z. Wang, Jia Li and Gio Wiederhold. "SIMPLicity: Semantics-Sensitive Integrated Matching for Picture Libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.23, no.9, pp.947-963, 2001.