

# 자기 조직화 맵 기반 유사 검색 시스템

오 창 윤<sup>†</sup> · 임 동 주<sup>††</sup> · 오 군 석<sup>†††</sup> · 배 상 현<sup>†††</sup>

## 요 약

특징 기반 유사성은 멀티미디어 데이터베이스 시스템에서 중요한 연구 쟁점이 되고 있다. 멀티미디어 데이터의 특징이 멀티미디어 객체들을 구별하는데 유용하다지만 특징 벡터의 차원의 수가 증가함에 따라 종래의 다차원 데이터 구조의 성능은 떨어지는 경향이 있다. R\*-Tree는 R-Tree의 가장 성공적인 변형으로 본 논문에서 고차원 특징 벡터를 위한 새로운 인덱싱 방법으로서 자기 조직화 맵 기반 R\*-Tree를 제안한다. 자기 조직화 맵 기반 R\*-Tree는 고차원 데이터를 좀더 스칼라화해서 탐색할 수 있도록 SOM과 R\*-Tree를 결합하여 구축한 인덱싱 기법이다. 자기 조직화 맵은 고차원 특징 벡터들로부터 2차원 공간으로의 맵핑을 제공한다. 그러한 맵을 위상 특징 맵이라 하고 인접 노드에 서로 유사한 특징 벡터들을 모아서 입력데이터의 특징 공간들 속에 유사성을 보존하는데 위상 특징 맵의 각 노드는 코드북 벡터를 가지고 있다. 실험적으로 4만개의 이미지로부터 추출된 색깔 특징 벡터들을 이용하여 자기 조직화 맵 기반 R\*-Tree의 검색시간 비용과 자기 조직화 맵과 R\*-Tree의 검색 시간 비용을 비교한다. 그 결과 R\*-Tree를 구축하는데 필요한 노드 수와 검색 시간 비용이 감소됨으로써 자기 조직화 맵 기반 R\*-Tree는 자기 조직화 맵과 R\*-Tree보다 훨씬 우수한 성능을 나타냄이 입증되었다.

## SOM-Based R\*-Tree for Similarity Retrieval

Chang-Youn Oh<sup>†</sup> · Dong-Ju Im<sup>††</sup> · Kun-Seok Oh<sup>†††</sup> · Sang-Hyun Bae<sup>†††</sup>

## ABSTRACT

Feature-based similarity has become an important research issue in multimedia database systems. The features of multimedia data are useful for discriminating between multimedia objects. The performance of conventional multidimensional data structures tends to deteriorate as the number of dimensions of feature vectors increases. The R\*-Tree is the most successful variant of the R-tree. In this paper, we propose a SOM-based R\*-Tree as a new indexing method for high-dimensional feature vectors. The SOM-based R\*-Tree combines SOM and R\*-Tree to achieve search performance more scalable to high-dimensionalities. Self-Organizing Maps (SOMs) provide mapping from high-dimensional feature vectors onto a two-dimensional space. The map is called a topological feature map, and preserves the mutual relationships (similarity) in the feature spaces of input data, clustering mutually similar feature vectors in neighboring nodes. Each node of the topological feature map holds a codebook vector. We experimentally compare the retrieval time cost of a SOM-based R\*-Tree with that of an SOM and R\*-Tree using color feature vectors extracted from 40,000 images. The results show that the SOM-based R\*-Tree outperforms both the SOM and R\*-Tree due to the reduction of the number of nodes required to build R\*-Tree and retrieval time cost.

**키워드:** 자기 조직화 맵(SOM), R\* 트리(R\*-Tree), 최적 매칭 이미지 리스트(BMIL), 최적 매칭 노드(BMN), 최소 경계 직사각형(MBR)

### 1. 서 론

다차원 데이터 집합을 다루기 위한 새로운 데이터베이스 애플리케이션의 사용이 증가함에 따라 이러한 데이터 집합으로 효과적인 질의 처리를 지원하는 기술이 중요한 연구 영역으로 간주되고 있다. 예를 들어 이미지 탐색의 경우 내용 기반 이미지 검색의 전형적인 질의가[1-3] “색상 이미지의 집합에서 유사한 색, 질감, 혹은 형상을 가진 이미지를 찾아라” 라는 것이라면 이 질의에서 사용된 특징들은 멀티미디어 객체들을 식별하는데 유용하다. 특징 벡터는 특

징들의 집합을 포함하는 벡터이고 보통 고차원 데이터로 표현된다. 그러한 고차원 특징 벡터들을 효과적으로 접근하기 위해 고차원 인덱스 트리라고 일컬어지는 많은 인덱싱 기술들이[4-7] 제안되고 있다.

본 논문에서 고차원 특징 벡터들을 위한 새로운 인덱싱 방법으로서 자기 조직화 맵(SOM: Self-Organizing Maps)[8, 9] 기반 R\*-Tree[5]를 제안하고 그러한 트리를 이용하여 이미지 데이터에 대한 유사성 탐색을 하고자 한다. 자기 조직화 맵은 신경회로망의 일종으로써 고차원 특징 벡터들의 데이터 집합으로부터 보통 2차원 공간으로의 맵핑을 제공하며 이러한 맵핑은 특징 벡터와 위상적 관계를 가지고 있어 위상 특징 맵이라 불리운다. 위상 특징 맵의 각 노드에 포함된 벡터들을 보통 코드북 벡터라 한다.

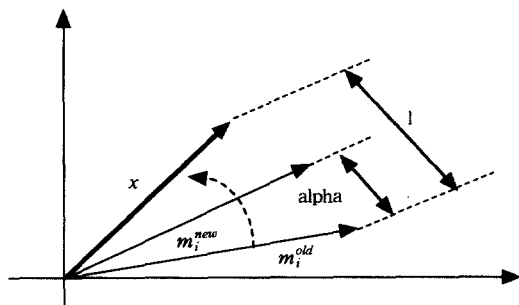
† 종신회원: 고창기능대학 정보통신설비과 교수  
 †† 정 회 원: 고창기능대학 정보통신설비과 교수  
 ††† 준 회 원: 큐수대학 시스템정보과학 연구과  
 †††† 종신회원: 조선대학교 전산통계학과 교수  
 논문접수: 2001년 4월 30일, 심사완료: 2001년 7월 16일

위에서 기술한 자기 조직화 맵의 이점에도 불구하고 학습된 자기 조직화 맵의 해석 가능성과 최적 매칭 노드에 대한 탐색시간의 복잡성이라는 결점 때문에 관련된 애플리케이션이 제약을 받고 있다. 최적 매칭 이미지 리스트인 BML(Best-Matching-Image-List)은 각 코드북 벡터에 가장 근접한 유사한 이미지들을 가지고 있다. 위상 특징 맵의 각 노드에는 둘 이상의 유사한 이미지가 분류되어 있는 반면에, 공노드에는 이미지가 분류되어 있지 않아 이러한 공노드는 불필요한 디스크 접근을 야기시켜 검색 성능을 저하시킨다. 그러므로 본 논문에서는 위상 특징 맵의 코드북 벡터들을 이용하여 R\*-Tree를 구축할 때, 공노드들을 미리 제거하고 자기 조직화 맵 속에 있는 노드의 수보다 더 적은 노드를 이용하여 인덱스를 구축할 수 있도록 한다. 따라서 유사 탐색 기술을 이용하면 탐색시간이 자기 조직화 맵의 탐색시간보다 더 빨라질 수가 있다.

2. 자기 조직화 맵 기반 R\*-Tree

2.1 자기 조직화 맵(SOM)

SOM은 위상적 관계를 가능한 한 충실하게 지키면서 고차원 입력 공간으로부터 보통 2차원 격자 상에 맵핑하는 신경회로망으로 2차원 격자에 배열되어 있는  $i$  노드의 집합으로 구성되어 있으며 각 노드에는 가중벡터  $m_i \in R^n$ 가 부가되어 있다.



(그림 1) 갱신된 자기조직화 맵(SOM)

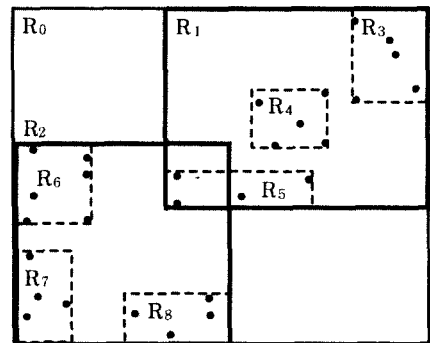
입력벡터  $x \in R^n$ 로 일컬어지고 있는 고차원 입력 공간으로부터의 요소들을 자기 조직화 맵에 주어지고, 입력벡터와 각 노드의 가중벡터 사이에 유클리드 거리를 계산하여 주어진 입력벡터에 대한 최적 매칭 노드(BMN: Best-Match-Node)를 구한다. 다음 단계에서 BMN의 가중벡터(가장 적은 유클리드 거리를 가지고 있는 노드)가 winner로 선택되어 주어진 입력벡터의 근사치로 수정되어진다. (그림 1)에서와 같이 학습시간의 변화에 따라서 감소되는 학습율  $\alpha(t)$ 에 따라 유클리드 거리의 어떤 한 부분만큼 winner의 가중벡터를 주어진 입력벡터 방향으로 이동시키는데, 여기서  $\alpha(t)$ 의  $t=1,2,3,\dots$ 은 불연속 시간 좌표로서 학습횟수를 나타내는 정수이고  $\alpha(t)$  ( $0 < \alpha(t) < 1$ )는 어느 정도의 크기로 입력벡터  $x$ 에 근접할 것인지를 조정하는 파라미터이

다. 만약  $\alpha=1$ 이면  $m_i$ 는 완전히  $x$ 와 일치하고  $\alpha(t)$ 는 1에 가까운 값에서 시작하여 지속적으로 감소한다. 따라서 이러한 BMN은 다음에 동일한 입력벡터가 주어질 때 훨씬 더 입력벡터에 근사한 값으로 수정된다. 또한 winner에 인접하는 노드의 가중벡터 역시 시간이 감소하는 인접함수  $\sigma(t)$ 에 의해 winner보다 정도는 작지만 갱신되어진다. 이러한 학습 절차는 결국 주어진 입력벡터들이 위상적으로 정렬된 맵핑을 가져오고, 유사한 입력 데이터는 맵 상의 인접 지역으로 맵핑된다[8]. 이러한 맵을 위상 특징 맵이라고 하고 위상 특징 맵 속의 노드가 가지고 있는 가중벡터는 코드북 벡터라 불리운다. 위상 특징 맵은 특징 공간 속의 입력 데이터의 유사성을 가지고 있고, 인접하는 노드들에 상호 유사한 특징 벡터들을 모은다. 학습단계 과정에서 입력벡터들은 유사한 입력벡터들이 서로 근접할 수 있도록 격자 상에 정렬되고 유사하지 않은 입력벡터들은 멀어진다. 이러한 이점에도 불구하고 자기 조직화 맵 애플리케이션은 한계가 있다. BMN에 대한 탐색은 자기 조직화 맵 알고리즘의 계산 시간을 지배하고 있으며, 고차원 입력 데이터나 대규모 자기 조직화 맵 네트워크일수록 계산 시간 비용이 많이 든다는 단점이 있다. 자기 조직화 맵의 기본 탐색 알고리즘은 완전탐색을 사용하는데, 이러한 탐색방법은 모든 노드를 탐색 대상으로 하기 때문에 탐색의 복잡성을 증가시키게 된다.

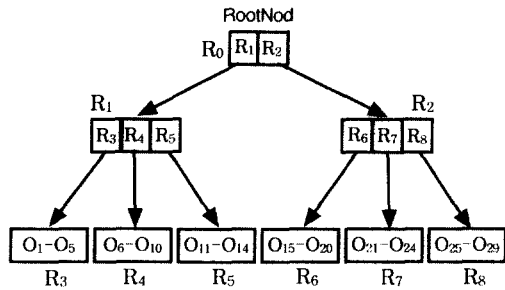
2.2 R\*-Tree

데이터베이스 시스템 분야에서 다차원 인덱싱 기술로서 다양한 인덱스 구조가 최근에 제안되었으며 이러한 기술 중에 하나로서 R\*-Tree는 R-Tree의 삼입과 분할 알고리즘을 수정하고 재삼입 메커니즘을 도입함으로써 R-Tree의 성능을 향상시키는 탐색 알고리즘이다[5]. R\*-Tree는 지리 데이터 또는 CAD 데이터와 같이 공간 데이터를 위한 인덱스 구조로서 제안되었으나 최근에는 이미지 검색을 위한 특징 공간의 인덱스 구조로서 사용되고 있다[10].

R-Tree의 가장 성공적인 변형인 R\*-Tree는 (그림 2)와 같이 직사각형 데이터를 위한 다차원 인덱스 구조이며 중첩된 직사각형의 계층구조와 높이가 균일한 트리이다. 모든



(a) 포인트 데이터의 공간



(b) 트리 접근 구조

(그림 2) R\*-tree 구조의 예

노드들은 계층구조에서 직사각형과 동일하고 각 노드에 대해서 디스크 페이지가 할당된다. 각 내부 노드는  $(p, \mu)$  엔트리의 배열을 포함하고 엔트리  $(p, \mu)$ 에서  $p$ 는 내부 노드의 자식노드에 대한 포인터,  $\mu$ 는 포인터  $p$ 가 포인팅하고 있는 자식노드의 최소 경계 직사각형(MBR : Minimum Bounding Rectangle)이다. 또한 각 단노드는 공간 객체에 대한  $(OID, \mu)$ 의 배열을 포함하고 있는데, 여기서  $OID$ 는 객체 식별자,  $\mu$ 는  $OID$ 가 구별하는 객체의 MBR이다. 따라서 루트노드의 직사각형은 모든 데이터 엔트리의 MBR과 일치하는 반면에 내부 노드의 직사각형은 하위 단노드에 포함되어 있는 데이터 엔트리의 MBR과 일치한다. R\*-Tree의 각 영역들은 서로 중복이 허용되어 동일 수준의 형제 영역들이 서로 중복될 수 있기 때문에 점질의(point query)에 대한 탐색시간은 트리의 높이가 아니라 중복의 양에 달려있다. 각 노드의 엔트리 수를 출력선의 수(fanout)라 하는데 루트노드의 fanout는 적어도 2이다. 다른 노드의 fanout는  $m$ 과  $M$ 사이인데  $m$ 과  $M$ 은  $2 < m < M/2$ 이다. 객체가 점 데이터일 때 단노드의  $M$ 값은 내부 노드의  $M$ 값과 다를 수 있다.

2.3 자기 조직화 맵 기반 R\*-Tree

자기 조직화 맵 기반 R\*-Tree의 구축을 위해 유사한 이미지들을 하나의 군집으로 묶는 것과 R\*-Tree를 구축하는 두 개의 프로세스로 구성하였다.

2.3.1 유사한 이미지들의 군집

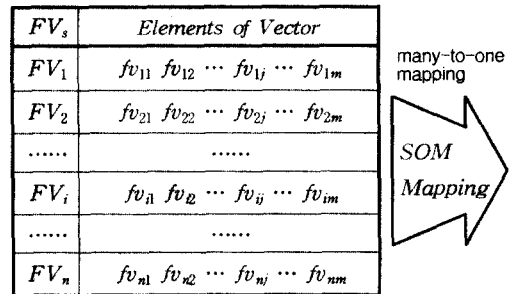
먼저 자기 조직화 맵을 사용하여 위상 특징 맵을 생성하고 위상 특징 맵으로부터 특징 벡터와 코드북 벡터 사이의 거리를 계산하여 BMIL을 생성한다. 위상 특징 맵을 생성하기 위해 학습 파라미터를 가지고 자기 조직화 맵 학습을 반복적으로 수행한다. 각 학습단계에서 특징 벡터는 입력 데이터 집합으로부터 연속적으로 선택되고 특징 벡터와 모든 가중 벡터사이의 거리가 계산되며 BMN을 맵 노드로부터 선택하여 가중벡터를 갱신한다. 그러므로 BMN과 위상적으로 인접한 노드들은 입력 특징 벡터에 더 근접하게 이동된다. 학습의 결과로서 맵의 각 노드 상에 생성되는 벡터는 코드북 벡터라 하고 다음과 같은 식으로 나타낼 수 있다.

$$CBV_i = [cv_{i1}, cv_{i2}, \dots, cv_{ij}, \dots, cv_{im}]^T$$

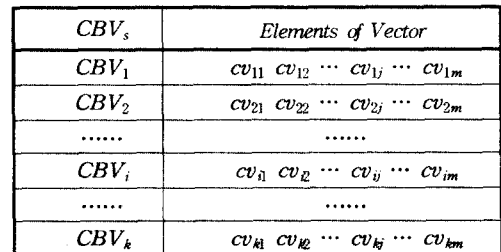
위 식에서  $i (1 \leq i \leq k)$ 는 맵의 노드 수,  $m$ 은 입력 노드 수로 특징 벡터의 차원을 의미하며  $k$ 는 맵 노드 수이다. 위상 특징 맵을 사용하여 가장 근접한 노드에 유사한 이미지를 분류시키는데 가장 근접한 노드는 주어진 특징 벡터와 모든 코드북 벡터 사이의 최소 거리를 가지고 있다. 각 노드의 분류된 유사한 이미지를 최적 매칭 이미지 리스트(BMIL)라 한다. 특징 벡터와 코드북 벡터사이의 유사성을 유클리드 거리에 의해 계산한다. 최적 매칭 노드  $BMN_i$ 는 다음과 같다.

$$BMN_i = \min(\|FV - CBV_i\|)$$

위 식에서  $FV$ 는 특징 벡터로 특징 벡터와 코드북 벡터사이의 관계는 (그림 3)에 나타나 있다. 이러한 두 종류의 벡터 사이에 다대일 관계가 형성되는데 이는 각 특징 벡터들 사이의 유사성에 기반을 두고 있다. 이상적인 맵의 형태는 이미지의 특징 벡터와 자기 조직화 맵의 각 노드가 가지는 코드북 벡터 사이에 일대일 대응이 되어야 하지만, 이는 일반적인 경우가 아니므로 어떤 맵 노드들은 다중 이미지들을 가지고 있다[11]. 이것이 의미하는 바는 BMIL이 생성될 때 위상 특징 맵 속에 공노드가 발생한다는 것으로 공노드는 매칭되는 어떠한 특징 벡터들을 포함하고 있지 않는 노드 공간의 일부분을 가리킴으로써 공노드 인덱싱은 불필요한 디스크 접근을 야기시켜 탐색 성능을 저하시킨다.



(a) 이미지로부터 추출한 특징벡터



(b) SOM에 의해 생성된 코드북 벡터

(그림 3) 특징 벡터와 코드북 벡터의 관계

그러므로 이를 위해 공간 요구사항은 공노드가 아닌 실노드만 인덱싱 함으로써 감소시킬 수 있다.

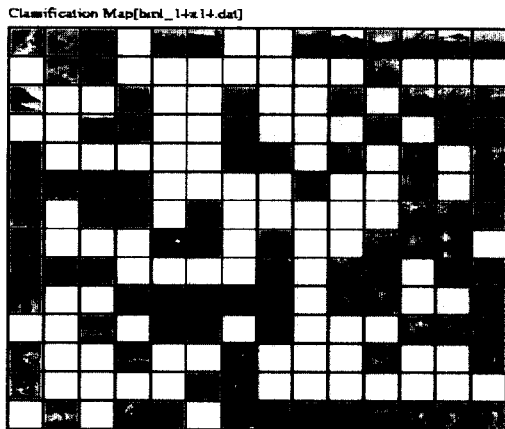
2.3.2 R\*-Tree의 구축

자기 조직화 맵 기반의 R\*-Tree를 구축하는데 R\*-Tree 알고리즘을 사용한다[5]. n차원 공간상에 하나의 점이 위상 특징 맵의 각 코드북 벡터와 일치한다고 했을 때 모든 코드북 벡터를 커버하는 공간은 루트노드와 일치한다. R\*-Tree를 구축하기 위해서는 위상 특징 맵으로부터 코드북 벡터를 선택하는데, 이것이 공노드이면 다음 코드북 벡터를 선택하고 그렇지 않으면 코드북 벡터를 삽입하는 단노드를 결정해야 한다. 새로운 엔트리를 삽입할 수 있는 가장 적합한 노드를 결정하기 위해서는 중심이 새로운 엔트리에 가장 가까운 서브트리를 선택하는 것이다. 노드나 단노드가 공간을 가지고 있을 때 엔트리가 추가되고 그렇지 않으면 재삽입이나 분할 알고리즘을 수행한다. 자기 조직화 맵 기반 R\*-Tree의 단노드는 다음 구조를 가지고 있다.

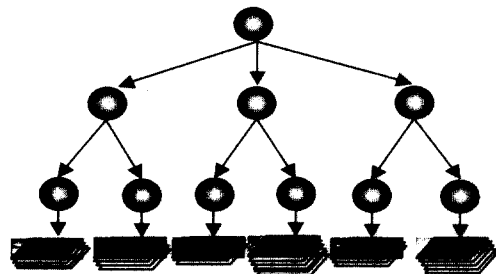
$$L : (E_1, \dots, E_i, \dots, E_p) \quad (m \leq p \leq M)$$

$$E_i : (OID, \mu)$$

단노드 L은 엔트리  $(E_1, \dots, E_i, \dots, E_p)$  ( $m \leq p \leq M$ )으로 구성되어 있는데 여기서 m과 M은 단노드 속에 있는 최소 그리고 최대 엔트리 수이다. 각 엔트리는 OID와 MBR  $\mu$ 를 포함하고 있다. 자기 조직화 맵 기반 R\*-Tree의 노드 구조는 (그림 4)에서 보여진 대로 R\*-Tree의 노드 구조와 동일하다.



(a) 위상 특징 맵의 예



(b) 공노드를 제거한 후 위상 특징 맵을 사용한 SOM을 기반의 R\*-tree

(그림 4) SOM 기반의 R\*-tree 구조

3. 시뮬레이션

자기 조직화 맵 기반 R\*-Tree를 일반적인 자기 조직화 맵과 R\*-Tree에 대하여 성능 비교 실험을 하였다. 실험에서 사용된 이미지 데이터베이스는 이미지 사이즈를 128x128 픽셀로 고정하여 4만개의 인공적/자연적 이미지로 구성된 풍경, 동물, 빌딩, 사람, 식물 등의 칼러 이미지를 포함하고 있다.

3.1 실험적 방법론

3.1.1 특징 추출

본 연구에서는 이미지 데이터로부터 칼러 특징을 추출해서 실험에 이용하였고 특징 벡터를 계산하기 위해 Haar wavelets[12]을 사용하였다. Haar wavelets은 wavelet 변형의 일종으로 가장 빠른 계산을 제공하고 실질적으로 잘 수행되는 걸로 알려졌다[1]. 하지만 Haar wavelets의 단점이 가장 중요한 하위 부분에서 이미지에 블록에러가 형성되는 경향이 있기는 하지만, 이는 유사성 검색에 현저히 영향을 주진 않는다[13]. 본 논문에서 특징 벡터를 위하여 사용된 칼러 공간은 luminance와 chrominance 정보를 가진 YIQ 공간이다[14]. Haar wavelets를 이용하여 3개의 칼러 공간의 각각에 대해 5 수준의 2차원 wavelet 변형을 계산하였으며 칼러 특징을 위해 가장 낮은 하위 행렬을 추출하여 특징 벡터를 생성하였다. 이러한 특징 벡터의 각 요소는 원래의 이미지의 평균 32x32 픽셀을 나타내고 칼러 특징 벡터는 3개의 채널로 된 YIQ 공간을 가진 48차원(4x4x3)으로 되어 있다.

3.1.2 자기 조직화 맵 기반 R\*-Tree의 구축

<표 1>에서 보여진 대로 맵 사이즈는 이미지 수와 거의 같다. 자기 조직화 맵의 학습을 통해서 칼러 특징 벡터를 사용하여 위상 특징 맵을 생성하였고 이러한 특징 맵을 사용하여 BMIL이 생성되었다. 공노드는 원래 맵 크기의 53~60%까지 점유하였으며 맵 크기가 커짐에 따라 공노드의 수도 증가한다. 이러한 공노드의 존재는 맵 크기와 상관없이 높은 유사성의 이미지가 동일한 노드 속에 분류된다는 것을 의미한다. 따라서 노드 수의 감소와 탐색시간의 단축은 공노드를 제거함으로써 실현될 수 있는데 이렇게 노드가 줄어든 집합으로 구축된 R\*-Tree는 전체적으로 좀 더 적은 인덱스 크기를 가지고 있다. <표 2>는 각각의 데이터 집합에 대해 일반적인 R\*-Tree의 구조와 자기 조직화 맵 기반 R\*-Tree의 구조를 비교하였다. 노드의 전체 수와 인덱스를 구축하는 시간 비용이 감소하더라도 트리의 높이는 그다지 큰 변화를 보이지 않았다. 이러한 관찰을 통해 메모리 사용과 검색 접근 시간이 감소된다는 것을 알 수 있고 데이터 집합이 크면 클수록 인덱스는 더욱 더 효율적이라는 것이 <표 3>을 통해서 명백히 입증되었다.

〈표 1〉 맵 사이즈와 공노드 표

Data set	Map size	Empty nodes	Ratio(%)
1000	32×32	551	53
5000	70×70	2,634	53
10000	100×100	5,395	54
20000	140×140	10,819	55
30000	175×175	18,377	60
40000	200×200	23,706	59

〈표 2〉 트리 구조

		Data set(×1000)					
		1	5	10	20	30	40
Total No. of nodes	R <sup>*</sup> -tree	119	499	972	2089	3042	3980
	SOM-based R <sup>*</sup> -tree	51	241	483	928	1300	1705
Height of tree	R <sup>*</sup> -tree	3	4	4	5	5	5
	SOM-based R <sup>*</sup> -tree	3	4	4	4	4	5
Time cost (sec)	R <sup>*</sup> -tree	7.55	50.44	111.89	233.89	350.66	476.49
	SOM-based R <sup>*</sup> -tree	3.27	19.12	40.76	81.95	115.49	153.92

〈표 3〉 R<sup>\*</sup>-tree 파일 크기

	Data set(×1000)					
	1	5	10	20	30	40
Original feature vector(MB)	1.38	5.78	11.27	24.22	35.26	46.13
SOM-based R <sup>*</sup> -tree(MB)	0.59	2.79	5.59	10.75	15.07	19.76
Ratio (%)	42	48	49	44	42	42

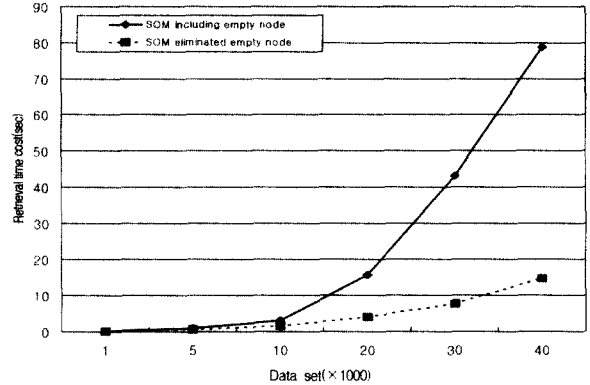
3.2 실험 결과

탐색시간을 측정하기 위해 다음과 같은 4가지 종류로 탐색 실험을 하였다.

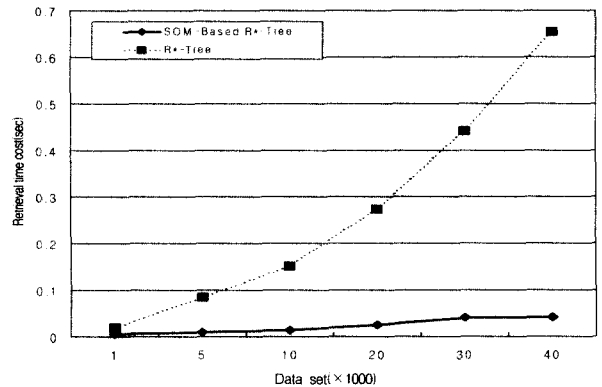
- 1) 공노드를 포함한 일반적인 자기 조직화 맵
- 2) 공노드를 제외한 일반적인 자기 조직화 맵
- 3) 일반적인 R<sup>\*</sup>-Tree
- 4) 공노드를 제외한 자기 조직화 맵 기반 R<sup>\*</sup>-Tree

데이터 집합 크기는 1,000~40,000개로 나누어진 6종류의 이미지 집합을 사용하였다. 사용된 탐색 방법은 k-Nearest Neighbor(NN)[15] 방법으로, 이 방법은 주어진 질의에 가장 가까운 k(k>1) 객체들을 탐색한다. 자기 조직화 맵에서 위상 특징 맵에 대한 완전한 탐색이 수행되고 주어진 질의에 가장 가까운 k(k=10) 노드를 찾는다. 이와 동일한 방법으로 일반적인 R<sup>\*</sup>-Tree와 자기 조직화 맵 기반 R<sup>\*</sup>-Tree는 k-MV(k=10) 탐색을 이용하는데 적용하였다. 가로축을 데이터 집합 크기로 하고 세로축을 검색시간 비용으로 한 비교가 (그림 5)와 (그림 6)에 나타나 있다. (그림 5)에서 보는 바와 같이 공노드를 가진 자기 조직화 맵의 검색시간은 공노드를 제거한 자기 조직화 맵과 비교할 때 데이터 집합 크기가 증가함에 따라 급격히 증가함을 알 수 있고 특히 40,000개의 이미지에서는 검색시간 비용이 5배가 더 들었다. 따라서 공노드를 제거하는 것이 불필요한 거리 계산을 없앨 수 있어 검색시간을 상당히 감소시킬 수 있음을 알 수

있었다. 또한 (그림 6)은 자기 조직화 맵 기반 R<sup>\*</sup>-Tree의 성능과 10-NN 검색시간 비용에 근거한 R<sup>\*</sup>-Tree의 성능을 비교하였으며 이러한 비교에서 주어진 질의에 대하여 가장 가까운 OID를 구할 수 있었다. 그러므로 자기 조직화 맵 기반 R<sup>\*</sup>-Tree의 검색시간은 R<sup>\*</sup>-Tree와 비교될 때 3배에서 15배까지 훨씬 더 짧음이 입증되었다.



(그림 5) 검색시간 코스트 그래프



(그림 6) SOM 기반의 R<sup>\*</sup>-tree와 R<sup>\*</sup>-tree 간의 검색시간 코스트 비교 그래프

이들 결과를 종합하여 볼 때, 원래의 특징 벡터 전체를 가지고 R<sup>\*</sup>-Tree를 구축하는 것이 검색 성능을 향상시킬 수 있었다. 더군다나 자기 조직화 맵 기반 R<sup>\*</sup>-Tree는 순차적으로 특징 벡터를 탐색하는 자기 조직화 맵보다는 성능이 훨씬 뛰어나며, 이러한 실험 결과로부터 자기 조직화 맵 기반 R<sup>\*</sup>-Tree가 유사성 검색에 대해 훨씬 효율적이라는 것을 명백히 보여준다.

4. 결 론

본 논문에서는 고차원 데이터 집합으로부터 유사성 검색을 다루는데 자기 조직화 맵 기반 R<sup>\*</sup>-Tree를 제안하였다. 자기 조직화 맵의 학습을 통해 얻어진 위상 특징 맵과 BMIL을 이용하여 R<sup>\*</sup>-Tree를 구축하였다. 본 연구 결과 위상 특징 맵에서 공노드를 제거한 R<sup>\*</sup>-Tree를 구축하는 것이 더 적은 노드를 가진 R<sup>\*</sup>-Tree를 생성하고 따라서 불필요한

접근이나, 노드 방문, 그리고 전체적인 탐색시간을 감소시킴으로써 성능을 향상시킨다는 것이다. 실험에서는 실제의 이미지 데이터를 이용하여 유사성 탐색을 수행하였고, 자기 조직화 맵 기반 R\*-Tree의 성능을 검색시간 비용이라는 측면에서 일반적인 자기 조직화 맵과 R\*-Tree의 성능을 비교하였다. k-NN 탐색을 이용한 실험 결과, 공노드를 제외하여 더 적은 노드로 구축한 자기 조직화 기반 R\*-Tree의 검색시간이 더욱 짧아진다는 것이 증명되었고, 자기 조직화 맵과 비교할 때 탐색 효율성이 매우 향상되었다.

**참 고 문 헌**

[1] A. F. C. E. Jacobs and D. H. Salesin, "Fast Multiresolution Image Querying," In Proc. SIGGRAPH95, pp.6-11, Aug. 1995.

[2] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber, "Efficient and Effective Query by Image Content," J. of Intell. Inform Syst., Vol.3, pp.231-262, 1994.

[3] V. N. Gudivada and V. V. Raghavan, "Content-based Image Retrieval system," IEEE Computer, Vol.28, No.9, pp.18-22, Sep. 1995.

[4] A. Guttman, "R-tree : a dynamic index structure for spatial searching," In proc. of ACM SIGMOD Int. Conf. on Management of Data, pp.45-57. 1984.

[5] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "R\*-tree : an efficient and robust access method for points and rectangles," In Proc. of ACM SIGMOD Int. Conf. on Management of Data, pp.322-331, May. 1990.

[6] S. Berchtold, C. Bohm, and H. P. Kriegel, "The pyramid-technique : towards breaking the curse of dimensionality," In Proc. of ACM SIGMOD int. conf. on Management of data, pp.142-153, June 1998.

[7] K. Chakrabarti and S. Mehrotra, "High dimensional feature indexing using hybrid trees," In Proc. of ICDE 1999, Mar. 1999.

[8] T. Kohonen. "Self-Organizing Maps," Springer, Berlin, 1997.

[9] T. Kohonen, "Self-organizing maps," Proc. of The IEEE, Vol.78, No.9, pp.1464-1480, 1990.

[10] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, j. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content : The QBIC System," IEEE Computer, Vol.28, No.9, pp.23-32, Sept. 1995.

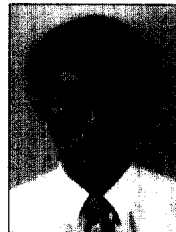
[11] M. Koskela. 'Content-Based Images Retrieval with Self-Organizing Maps,' Master's thesis, Helsinki Univ. of Technology, Dept. of Engineering Physics and Mathematics, 1999.

[12] S. G. Mallat, "Multifrequency Chnnel Decompositions of Images and Wavelet Models," IEEE. Trans., Acoust., Speech and Signal Proc., Vol.37, No.12, pp.2091-2110, Dec. 1989.

[13] A. Natsev, R. Rastogi, and K. Shim. "WALRUS : A Similarity Retrieval Algorithm for Image Databases," In Proc. ACM SIGMOD International Conference on Management of Data, pp.396-406, Jun. 1999.

[14] J. C. Russ, 'The Image Processing Handbook,' CRC Press, Boca Raton, 1995.

[15] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," In Proc. of the ACM SIGMOD, pp.71-79, May 1995.



**오 창 운**

e-mail : cssai@chollian.net

1992년 조선대학교 전산통계학과 (이학사)

1994년 조선대학교 전산통계학과 (이학석사)

1994년~1997년 (주)아시아자동차 근무

2000년 조선대학교 전산통계학과(이학박사)

2001년~현재 고창기능대학 정보통신설비과 교수

관심분야 : 컴퓨터 통신, 멀티미디어, 인공지능, 전자상거래, 지식베이스시스템



**임 동 주**

e-mail : imdongju@hanmail.net

1985년 전남대학교 영문학과(문학사)

1993년 미국 뉴욕주립대학교 전산학과 (이학석사)

1994년~1999년 대불대학교 근무

1999년 조선대학교 전산통계학과(이학박사)

2001년~현재 고창기능대학 정보통신설비과 교수

관심분야 : 컴퓨터네트워크, 멀티미디어, 소프트웨어엔지니어링, 데이터베이스 등



**오 군 석**

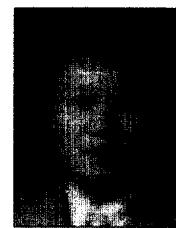
e-mail : okseok38@hotmail.com

1993년 일본 후쿠오카공업대학 관리공학과 (공학사)

1998년 일본 후쿠오카공업대학 정보공학전공 (공학석사)

2001년 큐수대학 시스템정보과학 연구과 (공학박사)

관심분야 : 화상 데이터베이스, 화상처리, 패턴인식, 컴퓨터네트워크, 멀티미디어 등



**배 상 현**

e-mail : shbae@chosun.ac.kr

1982년 조선대학교 전기공학과(공학사)

1984년 조선대학교 전기·전자공학과 (공학석사)

1988년 일본 동경도립대학 전자정보통신공학부(공학박사)

1995년~1996년 일본 NAIST 초빙교수

1988년~현재 조선대학교 전산통계학과 교수

관심분야 : 대규모 지식베이스, 인공지능경망, 퍼지시스템, GIS, 전문가시스템, 지식처리 등