

# 내용기반 검색을 위한 SOM $k$ -NN 탐색 알고리즘

## (SOM $k$ -NN Search Algorithm for Content-Based Retrieval)

오 군 석 \* 김 판 구 \*\*

(Kun Seok Oh) (Pan Koo Kim)

**요 약** 특징정보를 기반으로 한 유사 이미지 검색은 이미지 데이터베이스에 있어서 중요한 과제의 하나이다. 이미지 데이터의 특징정보는 각 이미지를 식별하는데 유용한 정보이다. 본 논문에서는 자기 조직화 맵 기반의 고속  $k$ -NN 탐색 알고리즘을 제안한다. 자기 조직화 맵은 고차원 특징벡터를 2차원 공간에 맵핑하여 위상특징 맵을 생성한다. 위상특징 맵은 입력 데이터의 특징공간과 상호관계(유사성)를 가지고 있으며, 인접노드에 서로 유사한 특징벡터가 클러스터링된다. 그러므로 위상특징 맵상의 각 노드에는 노드 벡터와 각 노드벡터에 가장 가까운 유사 이미지가 분류된다. 이러한 자기 조직화 맵에 의한 유사 이미지 분류결과에 대하여  $k$ -NN 탐색을 구현하기 위하여, (1) 위상특징 맵에 대한 접근방법, (2) 고속탐색을 위한 pruning strategy의 적용을 실현하였다. 본 연구에서는 실험을 통하여 실제 이미지로부터 추출한 색상 특징을 사용하여 제안한 알고리즘의 성능을 평가함으로써 유사 이미지 검색에 유효한 결과를 얻을 수 있었다.

**키워드** : 이미지 데이터베이스, 내용기반 검색, 자기조직화 맵, 최근접 탐색

**Abstract** Feature-based similarity retrieval become an important research issue in image database systems. The features of image data are useful to discrimination of images. In this paper, we propose the high speed  $k$ -Nearest Neighbor search algorithm based on Self-Organizing Maps. Self-Organizing Maps(SOM) provides a mapping from high dimensional feature vectors onto a two-dimensional space and generates a topological feature map. A topological feature map preserves the mutual relations (similarities) in feature spaces of input data, and clusters mutually similar feature vectors in a neighboring nodes. Therefore each node of the topological feature map holds a node vector and similar images that is closest to each node vector. We implemented a  $k$ -NN search for similar image classification as to (1) access to topological feature map, and (2) apply to pruning strategy of high speed search. We experiment on the performance of our algorithm using color feature vectors extracted from images. Promising results have been obtained in experiments.

**Key words** : Image databases, Content-based Image retrieval, self-Organizing maps, Nearest neighbor search

### 1. 서 론

최근, 멀티미디어 기술의 발달로 인하여, 대량의 이미지 데이터를 축적하여, 다양한 분야에서 활용할 수 있게 되었다. 그래서 그와 관련한 이미지 데이터베이스의 연구가 활발하게 진행되고 있다. 이미지 데이터베이스에

저장된 이미지를 효과적으로 활용하기 위해서는 사용자가 요구하는 이미지를 효율적으로 검색하기 위한 이미지 검색 기능을 갖추어야 할 필요가 있다. 이미지 검색을 실현하기 위해서는 크게 나누어, 이미지에 부여된 속성(텍스트에 의한 주석 등)을 이용하는 경우, 이미지 자체의 특징을 이용하는 경우 및 이러한 성질을 결합하여 이용하는 경우 등의 접근방식이 존재한다. 본 연구에서는 이미지 자체가 지니는 특징에 착안하여 내용 기반에 의한 이미지 검색(content-based image retrieval : CBIR) [3,7,21]에 관하여 논하기로 한다.

CBIR 접근 방식에서는 먼저, 데이터베이스에 등록된

\* 정 회 원 : 광주보건대학 디지털미디어과 전임강사  
okseok@www.kjhc.ac.kr

\*\* 종신회원 : 조선대학교 컴퓨터공학부 부교수  
pkkim@mina.chosun.ac.kr

논문접수 : 2001년 11월 5일

심사완료 : 2002년 8월 9일

이미지로부터 색상, 텍스처, 형상과 같은 이미지 고유의 시각적 특징정보를 추출한다. 특징정보는 이미지의 식별(discrimination), 즉 주어진 이미지가 어느 부류에 속하는가를 결정하는 데에 도움이 된다. 다음으로, 이러한 특징정보를 중심으로 각 이미지에 대하여 특징벡터를 생성하여, 각 이미지를 특징공간 상의 하나의 점에 대응시킨다. 이와 같은 전처리(preprocessing)에 의하여, 이미지 데이터베이스에 있어서의 유사검색은 특징공간의 질의에 대하여, 시스템이 제공하는 거리척도를 사용하여, 질의에 가장 가까운 순으로 이미지 데이터를 랭킹하는 최근접(nearest neighbor, NN) 탐색이 본질적인 처리가 되어 있다[8]. 이미지 데이터는 일반적으로 정보량이 반대하다는 것이 특징이다. 그러므로 대규모의 데이터베이스에서 주어진 질의 이미지에 대하여 유사한 이미지를 검색하는 데에는 실용적인 시간 내에 이루어져야 한다. 이와 같은 기술을 이용한 CBIR의 접근방식은 많은 시스템에서 이미 실증되었으며 [4,6,10,11,12], QBIC [4]과 같은 일부의 시스템은 상용화되어 있다.

본 연구에서는, 이미지의 특징정보를 유사성에 의하여 자동분류하기 위하여 Kohonen의 자기 조직화 맵(Self-Organizing Maps; SOM)[1,2]을 이용한다. SOM은 고차원 특징벡터를 가지는 데이터 집합을 2차원 상에 맵핑하여 특징벡터와의 위상적 관계(topological relationships)를 가지는 2차원 특징맵인 위상특징 맵을 생성한다. 이 위상 특징 맵의 각 노드(맵층의 유니트)에 생성되는 벡터를 일반적으로 코드북 벡터(codebook vector)라고 한다. 여기에서는 각 노드가 가지는 벡터라는 의미에서 노드 벡터라고 부르기로 한다. 이와 같이 SOM을 이용하여 특징벡터간의 유사성을 발견하여 상호간에 유사한 특징벡터끼리 맵상의 근접노드에 클러스터링할 수 있다. 그 결과 데이터베이스 내에는 서로 유사한 이미지가 포함되어 있기 때문에 일반적으로 1노드에는 2개 이상의 이미지가 맵핑된다. 반면 맵층의 노드에는 이미지가 하나도 분류되지 않은 공노드(empty node)가 생성된다. 이 공노드는 불필요한 디스크 접근을 야기하므로 전체의 검색성능을 저하시키는 요인으로 작용한다.

본 연구에서는 이와 같은 점을 고려하여 SOM에 있어서의 이미지의 유사검색을 고속으로 수행할 수 있는 k-NN 탐색 알고리즘을 제안한다. k-NN 탐색이란, 주어진 질의 이미지에 가장 유사한 이미지(실제로 거리척도에 의하여 질의 이미지에 가장 가까운 거리를 가지는 이미지)를 검색후보로서 k개를 탐색하여 검색결과로 하는 것으로, 특히 GIS(Geographic Information System)

에서 응용되어 왔다[8].

본 연구에서는 SOM을 기반으로 한 k-NN 탐색 알고리즘을 구현하기 위하여 다음과 같은 방법론을 제시한다. (1) 이미지의 특징정보를 학습하여 위상특징 맵을 생성하고, 위상특징 맵의 각 노드에 매칭되는 유사 이미지를 분류하여 최적매칭 이미지리스트를 생성한다. (2) 최적매칭 이미지리스트에 대하여 MBR(Minimum Bounding Rectangle)을 정의한다. 이러한 방법론에 의하여 k-NN 탐색알고리즘을 구현하여 그 유효성을 명확히 밝혔다.

본 논문의 구성은 다음과 같다. 2.에서는 관련연구에 대하여 알아보고, 3.에서는 유사 이미지의 분류에 대하여 논한다. 4.에서는 유사검색의 고속화를 위한 k-NN 알고리즘을 제안한다. 5.에서는 그 효율성을 검증하기 위한 실험과 제안된 알고리즘의 성능을 평가하였고, 6.에서 결론을 맺는다.

## 2. 관련연구

대용량의 멀티미디어 데이터베이스를 다루는데 있어서 주된 문제는 검색의 효율성으로, 이를 위한 적절한 색인기법을 설계해야 한다. 그 동안 다수의 다차원 색인기법들에 대한 연구가 진행되어 왔으나, R<sup>\*</sup>-tree[13]를 비롯한 색인기법들은 차원의 수가 증가함에 따라 시간 또는 공간 요구량이 지수적으로 증가되어 고차원의 경우 순차탐색보다 성능이 뒤지는 결과를 초래하여 색인 구조로서 효력을 상실하게 된다[14]. 차원의 증가에 따라 성능이 저하되는 문제를 “차원의 저주(the curse of dimensionality)”라고 한다[17,18]. 최근 이러한 문제를 해결하기 위하여 고차원 특징벡터를 수용할 수 있는 다수의 색인기법들이 연구되었다[13,14,15,16].

### 2.1 색인구조

R<sup>\*</sup>-Tree와 같은 다차원 색인구조는 많은 다른 다차원 색인구조와 비교할 만큼 성능이 좋은 것으로 잘 알려져 있다. 그러나 R<sup>\*</sup>-Tree의 성능은 차원수의 증가에 따라 성능이 저하된다[14]. 최근에 연구된 X-Tree[15]는 기존의 색인구조들이 차원이 증가함에 따라 검색 영역이 증가하여 검색성능이 현저히 저하되는 문제점을 방지하기 위해 제안된 고차원 색인구조이다. X-tree는 디렉토리의 검색 영역을 피하기 위한 분할 알고리즘과 수퍼노드 개념을 이용한다. X-tree구조는 노드를 분할할 경우 검색 영역이 최소가 되지 못할 때는 분할하지 않고 노드의 크기가 가변적으로 확장될 수 있는 수퍼노드를 사용한다. X-Tree구조는 저차원에서는 계층적인 디렉토리 구조를 사용하고, 고차원으로 갈수록 검색 영

역이 증가되기 때문에 기억공간이 절약되고 빠른 접근이 가능한 선형적인 디렉토리 구조를 사용한다. X-tree의 디렉토리 구조에 있어서 노드는 크게 MBR (Minimum Bounding Rectangle)정보를 갖는 중간 노드와 실제의 특징벡터 데이터를 갖는 데이터 노드로 구분된다. X-tree는 고차원 특징벡터 공간에서 기존의 색인구조에 비해  $k$ -NN 탐색질 의와 같이 유사성에 기반을 둔 검색을 효율적으로 지원하는 적절한 색인 구조로 평가된다.

A-Tree[16]는 VBR(Virtual Bounding Rectangle)과 상대셀(relative cell)의 개념, 그리고, 상대적 근사(relative approximation)에 관한 메커니즘을 주된 아이디어로 하고 있다. VBR은 MBR을 포함함과 동시에 그것에 근사한다. 또 상대셀은 데이터 객체를 포함하여 근사한다. A-Tree는 탐색에 있어서 부모 VBR의 상대적 위치에 기반을 두고 MBR과 객체가 근사한다. 이 상대적 근사에 의하여 A-Tree는 근사오차를 대폭적으로 억제할 수 있으며 균일한 분포를 가지는 데이터 집합뿐만 아니라, 불균형데이터 집합에 대해서도 높은 성능을 나타냈다. 그러나, A-Tree는 부모 VBR이 바뀌면 그 VBR을 참조하는 모든 MBR에 영향을 주므로, 동적인 환경에 있어서 높은 유지비용이 든다.

## 2.2 차원축소(dimensionality reduction)

$k$ -NN 탐색에 있어서 고차원의 영향을 줄이기 위해 차원의 축소(dimensionality reduction)[17]를 적용하여 트리구조에 데이터를 삽입한다. IBM의 QBIC시스템[4]은 이미지의 고차원 컬러 히스토그램(Color Histogram)을 3차원 공간의 하나의 점으로 맵핑하기 위하여 바운딩 기법을 사용한다. 이렇게 얻어진 공간상의 점을 이용하여  $R^*$ -Tree에 의한 색인구조를 구축한다. Chakrabarti 등[17]은 PCA(Principle component analysis)를 사용하여 LCR(Locally correlated cluster)간의 차원축소를 실행한다. 이러한 LCR은 하이브리드 트리구조를 이용하여 색인화 된다. 그러나, 이러한 시스템은 상호 관련된 클러스터를 찾아 차원을 줄이는 데이터 처리에 많은 비용이 든다. Faloutsos 등[6]은 DFT(Discrete Fourier Transform)을 사용하여 연속적인 시계열 데이터를 6차원 공간의 하나의 점으로 맵핑한다. 그리고 그 결과를 이용하여  $R^*$ -Tree를 사용하여 색인구조를 구축한다. QBIC의 바운딩 방법과 같은 기법은 색인구조에 질의점을 부여하여 데이터베이스의 후보집합을 검색한다. 그리고, 질의점과 각 후보간의 거리를 계산하여 검색결과를 정렬한다. 이 때 사용하는 거리계산은 상대적으로 많은 비용의 거리함수를 사용한다.

## 2.3 실용적인 $k$ -NN 탐색알고리즘

주어진 질의점과 가장 유사한 점  $k$ 개를 찾는 알고리즘이  $k$ -NN탐색 알고리즘이다. Roussopoulos 등[8]은 정확히 대응되는 점을 찾기 위하여 깊이 우선 탐색 알고리즘을 제안했다. 이 알고리즘에서는 다차원 검색공간에서 주어진 질의점으로부터 객체 또는 부검색 공간을 나타내는 최소경계사각형까지의 최소거리(minimum distance; MINDIST)와 최대탐색거리(minimax distance; MINMAXDIST)라는 개념을 이용하여 트리에서 검색되는 노드수를 줄이고자 하였다. Berchtold 등[20]은 각 데이터 포인트에 대응하는 보르노이 셀을 계산하여 최근접 탐색을 수행하는 접근방법을 제안했다. 그리고 보르노이 셀을 저장하기 위하여 X-Tree를 사용한다. 이 방법에서 사용하는 최근접 탐색은 질의점에 대응하는 보르노이 셀을 탐색하는 것이다. 이러한 접근방법은 전처리 비용이 많이 소요되며 X-tree의 효율성에 한계가 있는 것이다.

## 3. SOM에 의한 유사 이미지의 분류

본 연구에서는 이미지 데이터에서 Haar 웨이블릿 변환을 이용하여 각 이미지의 색상에 대한 특징정보를 추출하여 이미지의 유사검색에 이용하였다[9,10]. 색상을 표현하는 공간으로서는 YIQ공간[19]을 사용하여 색상 공간의 각 채널에 대하여 5레벨의 Haar 웨이블릿 변환을 수행하여 웨이블릿 계수를 구했다. 그 결과, 이미지 전체의 평균적인 색상 성분을 표현하는 가장 저대역 성분인  $4 \times 4$ 의 웨이블릿 계수를 추출하여, 48차원의 색상 특징벡터로 하였다.

유사 이미지의 분류는, 웨이블릿 변환에서 얻어진 색상의 특징벡터  $F_{col}$ 을 사용하여, SOM에 의한 학습을 수행한다. SOM은 2계층구조를 가지는 신경망(구조)이다(그림 1). 제1계층은  $n$ 차원의 입력계층이고, 그 노드 수  $n$ 은 특징벡터의 차원수와 동일한  $n=48$ 로 구성된다.

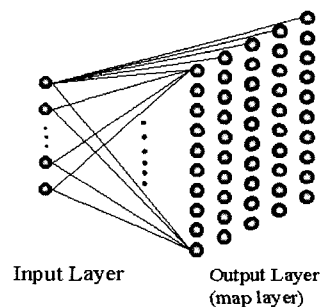
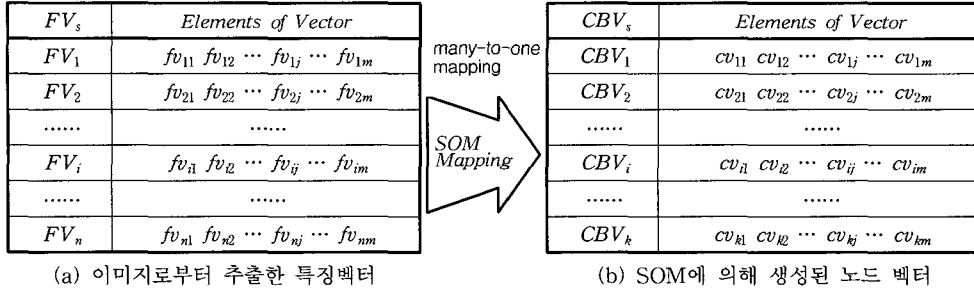


그림 1 자기조직화 맵의 구조



(a) 이미지로부터 추출한 특징벡터

(b) SOM에 의해 생성된 노드 벡터

그림 2 특징벡터와 노드 벡터와의 관계

제2계층은, 각 노드가 2차원으로 배열되어 있는 맵층(또는 경쟁층)으로 구성되고, 학습을 수행하기 전에 무작위의 가중치가 주어져 초기화된다. 유사 이미지 분류는, 아래와 같이 위상특징 맵의 생성과정과 최적매칭 이미지리스트 생성과정으로 이루어져 있다.

**3.1 위상특징맵의 생성과정**

위상 특징맵을 생성하기 위해서, 학습에 필요한 파라미터(학습률, 인접반경, 맵층의 사이즈, 인접함수, 학습회수)를 부여하여, SOM에 의한 학습을 수행한다.

위상특징 맵의 생성과정은 다음과 같다. 먼저, 입력 데이터로서 주어진 특징벡터에 가장 가까운 가중벡터를 맵층의 노드로부터 선택한다. 다음에, 그 노드와 인접영역내의 노드를 주어진 인접반경과 학습률 파라미터에 의하여 특징벡터에 접근하도록 갱신한다. 이 처리를 지정한 학습회수만큼 반복한다. 학습결과, 맵층의 각 노드에 생성되는 벡터를 노드벡터라 하며,

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

로 표현한다. 여기에서,  $i(1 \leq i \leq k)$ 는 맵층의 노드번호를 나타낸다.  $m$ 은 입력계층의 노드수(이미지의 특징벡터의 차원수)를,  $k$ 는 맵층의 노드수를 나타낸다. 이와 같이 얻어진 위상특징 맵은, 특징벡터의 상호간의 유사관계가 맵층의 2차원 상에 맵핑된 것이다.

**3.2 최적매칭 이미지리스트 생성과정**

최적매칭 이미지리스트(best-matching image list : BMIL) 생성과정에서는 위상특징 맵의 각 노드에 데이터베이스 내의 어느 이미지가 분류되는가를 계산하는 과정이다. 위상특징 맵을 이용하여 이미지의 특징벡터와 노드벡터간의 거리를 계산하여, 가장 가까운 노드(승자노드)를 구하여 그 노드의 유사 이미지로써 분류한다. 이 때 각 노드에 분류되는 유사 이미지가 가지는 정보는 이미지명과 노드벡터간의 거리이다. 특징벡터와 노드벡터간의 유사도는 유클리드 거리(Euclidean Distance)

에 의하여 계산했다. 승자노드  $BMN_i$ 는,

$$BMN_i = \min_{1 \leq i \leq k} \{ \| F_{col} - CBV_i \| \}$$

를 이용하여 구한다. 특징벡터와 노드벡터와의 관계를 (그림 2)에 보여준다. 이 2개의 벡터는, 거리척도에 의하여 다대일(many-to-one)의 대응관계에 있다.

이상의 처리를 데이터베이스 내의 모든 이미지에 대하여 색상 특징벡터를 이용하여 색상에 대한 위상특징 맵과 BMIL를 생성한다.

**4. SOM 기반 최근접 탐색의 고속화**

최근접(nearest neighbor) 탐색은, 주어진 질의점(query point)에 가장 가까운 점을 찾는 탐색방법이며, 이미지 등에서 얻어진 특징벡터를 사용하여 유사검색을 수행하는 경우 등, 멀티미디어 응용프로그램에 흔히 사용되는 기법이다[8].

본 연구에서는, SOM의 학습결과로 얻어진 위상특징 맵의 각 노드에 분류되어 있는 BMIL에 대하여, 기존의 공간 인덱스(spatial index)를 구축하는데 널리 사용되어 온 MBR(minimum bounding rectangle)[8]을 정의한다(4.2 참조). 그리고, 이 MBR에 대하여  $k$ -NN 탐색 알고리즘을 적용하여 효과적이고 고속의 유사 이미지 검색을 실현하고자 한다.

**4.1 승자노드를 중심으로 한 인접영역에 있어서의 탐색**

$k$ -NN 탐색에서는 최초로 방문하는 노드를 결정할 필요가 있다. 주어진 질의점과 가장 가까운 거리를 가지는 탐색영역이 그 대상이 된다. 위상특징 맵에서는, 주어진 특징벡터와 맵상의 각 노드가 가지는 노드벡터간의 거리를 계산하여 최소가 되는 노드를 승자노드(winner node)라고 한다. 본 연구에서는 이 노드를 제1 방문노드로 했다. 또 SOM은 승자노드의 인접노드가 서로 유사하다는 성질을 가지고 있다. 따라서, 승자노드를 중심으로 한 인접노드에 대하여 탐색을 진행하는 것이 질의 이

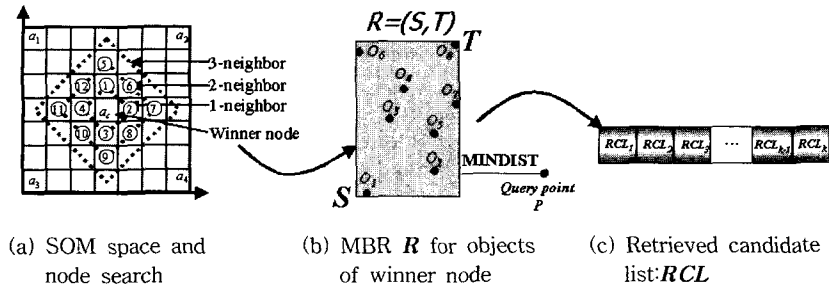


그림 3 승자노드를 중심으로 한 k-NN 탐색

미지에 보다 더 가까운 이미지를 고속으로 검색하는 데 유효한 방법이다. 맵층의 인접영역을 (그림 3)의 (a)와 같이 정의한다. 인접영역은 승자노드와 그에 인접하는 노드간의 거리가  $L_1$ 공간의 도시블럭거리(City Block Distance)[1]에 의하여 결정된다. 탐색시에는, (그림 3)의 (a)와 같이 인접노드의 반경을 확대해 가면서 주어진 질의 이미지에 유사한 이미지를 순차적으로 검색한다. 맵층에서 탐색해야 할 최대 인접영역의 반경은,

$$Y = \max_{1 \leq i \leq 4} (\| a_c - a_{i1} \|)$$

로 정의한다. 여기에서,  $a_c$ 는 승자노드의 위치벡터,  $a_i$ 는 격자상의 4점의 위치벡터를 나타낸다.

4.2 각 노드의 MBR 정의

맵상의 각 노드에 대한 MBR  $R$ 을 정의하기 위하여, 3. 에서 생성한 BMIL을 사용한다. BMIL은 각 노드에 분류된 객체  $O = \{ o_{ij} | 1 \leq i \leq N, 1 \leq j \leq n \}$ 로 구성되어 있으며, 상호간에 유사성을 포함하고 있다. 여기에서,  $N$ 은 노드에 분류된 객체수,  $n$ 은 각 객체가 가지는 특징벡터의 요소를 나타낸다. 또한 본 논문에서 객체란 이미지 데이터를 의미한다. MBR  $R$ 은 (그림 3)의 (b)와 같이,

$$R = (S, T)$$

로 정의한다. 여기에서,  $S = \{ s_j | 1 \leq j \leq n \}$ ,  $T = \{ t_j | 1 \leq j \leq n \}$ 은 각각 MBR  $R$ 에 있어서 최소가 되는 점과 최대가 되는 점을 나타내며,

$$s_j = \min_{1 \leq i \leq N} (o_{ij}) \quad j = 1, 2, 3, \dots, n$$

$$t_j = \max_{1 \leq i \leq N} (o_{ij}) \quad j = 1, 2, 3, \dots, n$$

에 의하여 계산한다. 여기서  $s_j \leq t_j$ 이다. 이 MBR  $R$ 의 계산은 공노드를 제외한 맵층의 모든 노드에 대하여 수행했다. 위상특정 맵의 MBR은 상호간에 중복되지 않는다. 그 이유는 각 노드에 이미 유사 이미지가 분류되어 있으며, 각 노드간에 중복되는 이미지를 갖고 있지 않기 때문이다.

다음에, 주어진 질의점  $P$ 와 MBR  $R$  과의 최단거리

MINDIST는

$$MINDIST(P, R) = \left( \sum_{i=1}^n | p_i - r_i | \right)^{1/2}$$

로 정의한다. 여기서,

$$r_i = \begin{cases} s_i & \text{if } p_i < s_i \\ t_i & \text{if } p_i > t_i \\ p_i & \text{otherwise} \end{cases}$$

이다. 즉, (그림 3)의 (b)와 같이, MINDIST는 MBR속의 모든 점과 질의점  $P$ 와의 거리가 최소가 되는 점의 거리를 의미한다. MINDIST의 거리계산은, 특징벡터간의 거리척도이므로,  $L_2$ 공간의 유클리드 거리를 이용한다. 이 최단거리  $MINDIST(P, R)$ 은 탐색시에 탐색이 불필요한 노드의 판단에 이용된다.

4.3 Pruning Strategy

대규모의 데이터를 실용적인 시간 내에 탐색하기 위해서는 최근접 탐색의 고속화가 불가결하다고 생각한다. 그러기 위해서는 모든 데이터에 대한 탐색보다는, 탐색이 불필요한 데이터를 pruning 처리함으로써 탐색시간의 고속화를 실현할 수 있다.

본 논문에서는, 탐색이 불필요한 노드를 판단하기 위해서 다음과 같은 pruning strategy를 결정한다.

- (i) 맵의 노드중에서 객체(이미지)를 하나도 가지지 않는 공노드를 탐색대상에서 제외한다.
- (ii)  $MINDIST(P, R) < D(RCL_k)$ 이면 탐색대상의 노드이며, 그렇지 않으면 탐색대상에서 제외된다.

여기에서 RCL이란, 검색후보 리스트(retrieved candidate list)를 말하며, (그림 3)의 (c)와 같이 질의점에 가장 가까운 노드(승자노드)로부터 탐색영역을 인접영역으로 확대해 가면서 얻어진  $k$ 개의 검색결과를 저장하고 있는 리스트이다. 그러므로 검색후보 리스트 RCL에 있어서  $RCL_1$ 은 항상 질의점과 가장 가까운(또는 일치하는) 거리에 있으며,  $RCL_k$ 는 질의점으로부터 가장 먼 거리에 있는 검색후보를 나타낸다. 따라서  $RCL_k$

는 탐색대상을 판단하는 기준이 된다. 이와 같이 탐색대상의 pruning 처리에 의하여 탐색시간을 절감할 수 있다.

**4.4 SOM 기반 k-NN 탐색 알고리즘**

본 연구에서 제안하는 k-NN 탐색 알고리즘은 2개의 Phase로 이루어져 있다. 탐색 알고리즘의 의사코드(pseudo code)를 (그림 4)과 (그림 5)에 나타낸다.

```

Phase 1: ComputeWinnerNode
Input : q_Object
Output : WinnerNode, retrievedList
1: Nearest = ∞;
2: qf : feature vector of q_Object
3: i: node number on the Map
4: forall Node on the Map do begin
5:   Disti = ||cbvi - qf||;
6:   if( Nearest > Disti ) then
7:     do begin
8:       Nearest = Disti;
9:       WinnerNode = i;
10:    end
11:   i = i + 1;
12: end
13: /* initialize k-retrieved Image List */
14: /* from WinnerNode */
15: update RCL with WinnerNode
    
```

그림 4 Phase-1: 승자노드 탐색과 검색후보 RCL의 초기화

```

Phase 2: SearchNearestNeighbor
Input : γ, qf, k_NN
Output : retrievedList
1: γ: current NN radius
2: γ: maximum of NN radius
3: qf: feature vector of q_Object
4: R = (S,T); /* MBR */
5: i: node number of the Map
6: if( γ > γ ) then do begin
7:   return
8: else
9:   get NNodeList; /*Neighbor Node List*/
10:  for(NNodeList, ∈ γ) do begin
11:    /* prune empty Node */
12:    if(NNodeListi is emptyNode) then
13:      do begin
14:        discard NNodeListi;
15:      end
16:    /* prune visiting unnecessary Node */
17:    Dist = MINDIST(qf, R);
18:    if(Dist > RCL[k_NN].Dist)
19:      then do begin
20:        discard visiting unnecessary Node
21:      end
22:    update RCL with NNodeListi;
23:    i=i+1;
24:  end
25:  γ = γ + 1;
26:  call SearchNearestNeighbor( γ );
27: end
    
```

그림 5 Phase-2: γ 영역에 있어서의 유사검색 알고리즘

Phase-1(그림 4)에서는 먼저 위상특징 맵으로부터 주어진 질의 이미지에 가장 가까운 승자노드를 구한다(Step 4-12), 이 승자노드에 분류되어 있는 이미지 중에 질의 이미지에 가장 유사한 k(k>0)개를 최초의 검색후보로서 검색 후보리스트(RCL)에 정렬하여 저장한다(Step 15). 검색 후보리스트는 질의 이미지에 가장 가까운 후보로부터 항상 정렬되어 저장되기 때문에, 리스트 속의 k번째의 후보는 검색결과 중에서 질의 이미지로부터 가장 거리가 먼 후보를 유지하고 있다. 또한 이 처리에서는 맵층의 탐색해야 할 최대 인접영역 γ가 얻어진다.

Phase-2(그림 5)에서는, 승자노드를 중심으로 인접영역의 반경 γ를 확대해 가면서 최근접 탐색을 수행한다. 이 처리에서는 탐색시에 앞에서 논한 바와 같이 2가지의 pruning strategy를 적용한다(Step 11-21). 그 결과 탐색대상의 노드에 분류되어 있는 이미지를 가지고 검색 후보 리스트를 갱신한다(Step 22). 이 처리는 γ ≤ γ 인 동안에 재귀적으로 실행된다(Step 26).

**5. 실험결과와 고찰**

SOM에 있어서의 최근접 탐색 알고리즘의 유효성을 검증하기 위하여 제안한 k-NN 알고리즘을 구현하여, 이미지의 유사검색을 예로서 성능평가 실험을 수행했다. 실험에 사용한 이미지 데이터는 스탠포드 대학에서 공개되어 있는 이미지 컬렉션<sup>1)</sup>과 H<sup>2</sup>soft<sup>2)</sup>의 이미지 데이터를 합하여 40,000 이미지로 구성되어 있다. 이미지 데이터베이스의 사이즈는 <표 1>(Data set)에 나타낸 바와 같이, 크기가 서로 다른 6종류의 데이터 집합을 작성하여 사용하였다. 그리고, 각 이미지 데이터의 사이즈는 128×128 픽셀로 고정했다. 각 이미지의 특징벡터는 3.에서 기술한 48차원의 색상에 대한 특징벡터를 사용했다.

표 1 각 데이터 집합에 대한 맵 사이즈와 공노드

Data set	Map size	Empty nodes	(%)
1000	32×32	434	42
5000	70×70	2634	53
10000	100×100	5395	54
20000	140×140	10819	55
30000	175×175	18377	60
40000	200×200	23706	59

1) <http://WWW-DB.Stanford.EDU/IMAGE/>  
 2) <http://www.h2soft.co.jp>

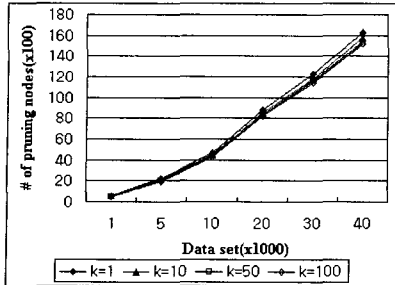


그림 6 Pruning strategy에 의한 탐색이 불필요한 노드수(공노드를 제외)

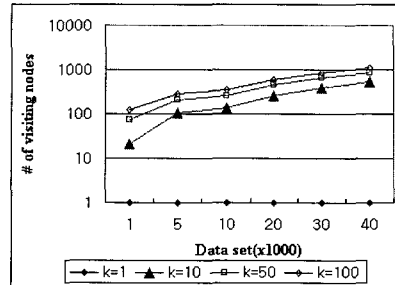


그림 7 k 값의 증가에 따른 탐색 노드수

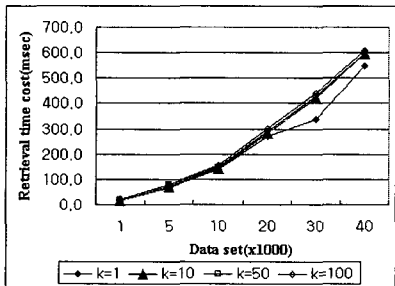


그림 8 k 값의 변화에 따른 전체 검색시간

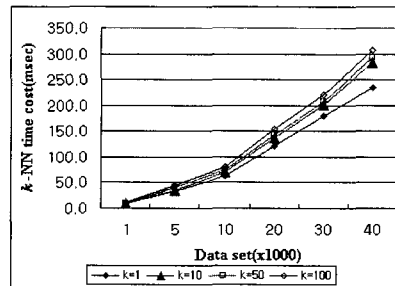


그림 9 k-NN 탐색시간

각 데이터베이스에 대하여 무작위로 선택한 20개의 질의 이미지에 대하여 이미지 검색의 성능평가 실험을 수행하고, 그 평균치를 측정결과로 했다. 프로그램 언어는 C++를 사용하여 SUN Microsystems사의 Ultra5 (CPU: UltraSPARC-IIi, OS: Solaris8, Memory: 512MB)상에서 측정했다. 이미지 데이터의 색상에 대한 특징벡터를 입력 데이터로 하여 SOM에 의한 학습을 수행했다. SOM의 맵층의 사이즈는 <표 1>(Map Size)에 나타난 바와 같이 각 데이터 집합의 이미지 수와 거의 동일하게 설정했다. 맵층의 각 노드가 가지는 가중벡터는 특징벡터의 차원수와 같도록 무작위로 초기화된 48차원의 벡터이다. 학습회수는 T=10,000으로 하고, 학습과정의 파라미터로서 인접집합의 초기치를  $\sigma_0=30$ , 학습률의 초기치를  $\alpha_0=0.9$ 로 했다.

인접집합과 학습률은 학습이 수행되는 동안에 계속해서 갱신되는 감소함수이다. SOM의 학습결과 맵층의 각 노드에는 일반적으로 1노드에 여러 개의 이미지가 맵핑된다. 그 이유는 데이터베이스 내에는 서로 유사한 이미지가 포함되어 있기 때문이다. 따라서, 맵층의 노드수를 데이터 집합과 거의 동등하게 설정하면 맵층의 노드에는 이미지가 하나도 분류되지 않은 공노드가 생긴다. 본 실

험에서 학습을 수행한 6종류의 데이터 집합으로부터 얻어진 맵층의 공노드의 비율을 <표 1>(Empty nodes, (%))에 나타낸다. 이 공노드를 제외하고 맵층의 각 노드에 대하여 MBR을 구했다.

주어진 질의 이미지에 대하여 k-NN검색을 수행한다. 실험에서는 검색후보 k의 값을 8종류(k=1, 5, 10, 20, 30, 40, 50, 100)에 대하여 검색실험을 했다. 본 실험에서는 데이터베이스 내의 이미지 데이터를 질의 이미지로 사용했다. 그러므로 k=1인 경우의 검색결과는 질의 이미지와 동일하다(그림 7).

질의 이미지와 데이터베이스 내의 이미지와의 거리측도는  $L_2$  공간의 유클리드 거리를 사용한다. 실험은 k의 값과 데이터 집합사이의 사이즈의 증가에 따라서 k-NN 탐색 알고리즘의 동작과 MINIDIST의 영향에 대하여 검토하였다.

(그림 6)은 각 데이터 집합에 대하여 k의 증가에 따라 탐색할 필요가 없는 노드수(pruning nodes)를 나타낸 것이다. 이 측정결과에서는 공노드의 수는 포함되어 있지 않다. 그 이유는 공노드수는 이미 알고 있는 값이며, 여기에서는 MINDIST의 영향을 조사하는데 그 목적이 있기 때문이다. (그림 6)과 (그림 7)을 비교해 보

면, 검색후보  $k$ 가 증가함에 따라 탐색할 방문의 노드수도 증가하지만, pruning 노드수의 비율은 전체 노드수의 38%(Data set=40,000,  $k=100$ )에서 57%(Data set=1000,  $k=1$ )로 나타났다. 이것은 MINDIST에 의한 pruning strategy 효과가 작용했다고 볼 수 있다. 또 (그림 7)의  $k$ 의 증가에 대한 탐색대상 노드수를 보면  $k=1$ 일 때는, 데이터 집합 40,000이외에서는 승자노드만이 탐색대상이 되었다.  $k=100$ 의 경우는 데이터 집합 40,000에서는 전체 노드수의 2.7%에 해당하는 1084개의 노드에 접근했다. 이 두가지 실험에 의하여 SOM에 있어서  $k$ -NN 탐색을 수행하기 위하여 정의한 pruning strategy의 유효성을 명확하게 밝혔다.

$k$ -NN 탐색 알고리즘의 검색시간을 (그림 8)에 나타낸다. 여기에서 검색시간이란, 질의 이미지에 대하여 승자노드의 결정과 인접영역에 대한  $k$ -NN 탐색시간, 그리고 탐색노드에 대하여  $k$ 개의 검색결과를 얻는데 소요된 시간을 말한다. (그림 9)는 승자노드의 인접영역에 대한  $k$ -NN 탐색시간만을 나타낸 것이다.

이상과 같이 실험결과를 통하여  $k$ 의 값이 증가해도 검색시간은 큰 폭으로 변하지 않는다는 점을 들 수가 있다. 그 이유는 (i) 검색후보  $k$ 가 증가해도 탐색영역은 승자노드를 중심으로 한 2차원 맵에 한정되므로 탐색해야 할 인접영역은 변하지 않는다는 점(4.1의 최대 인접영역의 정의), (ii) 방문한 각 노드에 분류되어 있는 이미지와 질의 이미지와의 거리계산이 메모리 상에서 수행되므로 그 계산시간은 거의 무시된다는 점을 들 수가 있다.

## 6. 결론

본 연구에서는, SOM의 학습결과에서 이미지의 유사 검색을 고속으로 수행하는  $k$ -NN 탐색 알고리즘을 제안했다. 또 고속탐색 알고리즘을 구현하기 위하여, 맵층의 각 노드에 대한 MBR의 적용, 주어진 질의점과 MBR과의 최단거리함수 MINDIST를 정의하였다.

우리는 제안한 알고리즘을 구현하여, 실제의 이미지 데이터를 사용하여 유사 이미지 검색의 실험을 수행하여 그 유효성에 대하여 평가했다. 이번 실험을 통하여 MINDIST에 의한 유사 이미지 검색의 고속화가 실현된 것이 증명되었다. 또 검색시간은 검색후보  $k$ 의 증가에 관계없이 거의 차이를 보이지 않았다는 점에 그 의의가 있으며 SOM에 의한  $k$ -NN탐색과 pruning strategy의 유효성이 검증되었다. 그리고 예제 이미지에 의한 검색 기법을 사용해도 실용적인 시간내에 기대되는 유사 이미지를 검색하는 것이 가능하게 되었다.

향후 과제로서, 검색후보  $k$ 의 수에 대하여 접근하는 이미

지수가 너무 많아 검색시간에 영향을 준다는 점을 고려하여 알고리즘을 개선해 나갈 예정이다. 또 실제의 응용프로그램으로서 구현하여 동작확인고 성능평가를 할 예정이다.

## 참고 문헌

- [1] Kohonen, T., "Self-Organizing Maps," Series in Information Sciences, vol.30, Springer-Verlag, Third edition, Berlin (2001).
- [2] Kohonen, T., Hynninen, J., and Laaksonen, J., "SOM\_PAK: The Self-Organizing Map Program Package," In Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information science (1996).
- [3] Gudivada, V. N. and Raghavan, V. V. eds., "Content-based Image Retrieval System," IEEE Computer, Vol.28, No.9, pp. 18-22 (1995).
- [4] Flickner, M. et al., "Query by Image and Video Content: The QBIC System," IEEE Computer, Vol.28, No.9, pp. 23-32 (1995).
- [5] Faloutsos, C. et al., "Efficient and Effective Query by Image Content," J. Intell. Inform. Syst., Vol.3, pp.231-262 (1994).
- [6] Faloutsos, C., Ranganathan, M., and Manolopoulos, Y., "Fast Subsequence Matching in Time-Series Databases," ACM SIGMOD, 1994.
- [7] Rui, Y., Huang, T. and Chang S-F., "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," J. Visual Communication and Image Representation(JVCIR), Vol.10, No.1, pp. 39-62 (1999).
- [8] Roussopoulos, N., Kelley, S. and Vincent, F., "Nearest neighbor queries," In Proceedings of the ACM SIGMOD Conference, pp.71-79, San Jose, CA, May (1995).
- [9] Mallat, S. G., "Multifrequency channel decompositions of images and wavelet models," IEEE. Trans., Acoust., Speech and Signal Proc., Vol.37, No.12, pp. 2091-2110 (1989).
- [10] Smith, J. R. and Chang, S-F., "VisualSEEK: A Fully Automated Content-Based Image Query System," Proc. ACM Intl. Conf. on Multimedia, Boston, pp.87-98 (1996).
- [11] Pentland, A., Picard, R. W. and Schlaroff, S., "Photobook: Tools for Content-Based Manipulation of Image Databases," SPIE Storage and Retrieval for Image and Video Databases II, San Jose, pp. 34-47 (1994).
- [12] Jacobs, C. E., Finkelstein, A. and Salesin, D. H., "Fast multiresolution Image Querying," Proc. SIGGRAPH95, Los Angeles, California, pp.6-11



- (1995).
- [13] 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.
  - [14] Chakrabarti, K., and Mehrotra, S., "The Hybrid Tree: An Index Structure for High Dimensional Feature Spaces," In Proc. of IEEE ICDE99, Mar. 1999.
  - [15] Berchtold, S., Keim, D., and Kriegel, H. P., "The X-Tree: An Index Structure for High-Dimensional Data," VLDB, 1996.
  - [16] Sakurai, A., Yoshikawa, M., and Uemura, S., "The A-Tree: An Index Structure for High-Dimensional Spaces Using Relative Approximation," VLDB2000, 2000.
  - [17] Chakrabarti, K., and Mehrotra, S., "Local Dimensionality Reduction: a New Approach to Indexing High Dimensional Spaces," VLDB2000, Egypt, 2000.
  - [18] Bellman, R., "Adaptive Control Process: A Guided Tour," Princeton University Press, pp.94-95, 1961.
  - [19] Russ, J. C., "The Image Processing Handbook," CRC Press, Boca Raton, 1995.
  - [20] Berchtold, S., Ertl, B., Kriegel, H. P., and Seidl, T., "Fast Nearest Neighbor Search in High-Dimensional Space," Int'l Conf. on Data Eng. (ICDE), 1998.
  - [21] Smeulders, A., Worring, M., Santini, S., Gupta, A., and Jain, R., "Image Databases at the end of the Early Years" IEEE Transactions on Pattern Analysis and Machine Intelligence Vol.23, No.1, January 2001.



오 군 석

1993년 후쿠오카공업대학 공학부 관리공학과(공학사). 1998년 동대학 대학원 공학연구과(공학수사). 2001년 큐슈대학 대학원 시스템정보과학연구과(공학박사). 2002년 ~ 현재 광주보건대학 정보기술계열 디지털미디어과 교수. 관심분야는 이미지 데이터베이스, 이미지처리, 패턴인식



김 판 구

1988년 조선대학교 컴퓨터공학과(공학사). 1990년 서울대학교 컴퓨터공학과(공학석사). 1994년 서울대학교 컴퓨터공학과(공학박사). 1995년 ~ 현재 조선대학교 컴퓨터공학부 교수. 관심분야는 시스템 보안, 운영체제, 정보검색, 영상처리