

논문 2006-43CI-1-7

# 클러스터링과 차원축약 기법을 통합한 영상 검색 시스템

## ( Combined Image Retrieval System using Clustering and Condensation Method )

이 세 한\*, 조 정 원\*\*, 최 병 욱\*\*\*

(Se-Han Lee, Jungwon Cho, and Byung-Uk Choi)

## 요 약

본 논문에서는 전체 차원으로 데이터베이스 내의 모든 영상에 대해 순차적인 검색을 했을 때의 상세 검색 결과와 동일한 적합성을 유지하면서 검색 속도를 훨씬 더 향상시킬 수 있는 통합 검색 시스템을 제안한다. 통합 검색 시스템은 적합성을 유지하는 서로 다른 두 독립적인 시스템이 병합되어 있다. 하나는 특징 벡터 차원 축약을 이용한 2단계 검색 시스템이고, 나머지 하나는 이진 트리 클러스터링을 이용한 2단계 검색 시스템이다. 각각의 방법은 1단계에서 상세 검색에서의 검색 결과를 항상 포함하는 후보 영상들을 추출하고, 추출된 후보 영상들을 대상으로 2단계 검색에서 전체 차원으로 재 검색을 한다. 그러므로 각 방법과 통합 검색 방법은 모두 상세 검색을 수행했을 때와 동일한 검색 결과를 얻게 된다. 특징 벡터 차원 축약을 이용한 2단계 검색 방법은 Cauchy-Schwartz 부등식의 성질을 이용하여 특징 벡터를 차원 축약하여 검색에 사용하는 방법이다. 이때 전체 검색 시간을 최소화 하는 최적 차원 축약율이 존재하게 되고, 이를 후보 영상 추출을 위한 1차 검색에 적용하게 된다. 이진 트리 클러스터링을 이용한 2단계 검색 방법은 재귀적인 2-means 클러스터링을 통해 각 클러스터의 반경이 동일하게 동적으로 분할하는 방법이다. 동일한 적합성 유지를 위해 유사도 기준이 보정된 질의를 통해 1단계 검색에서 후보 클러스터를 추출하고, 2단계 검색에서 후보 클러스터 내의 영상을 대상으로 최종 결과 영상들을 얻게 된다. 통합 검색 방법은 위의 두 검색 방법을 통합한 것으로 서로 독립적인 두 방법을 동시에 적용함으로써 검색 시스템의 성능을 훨씬 더 향상시킬 수 있다. 제안하는 방법은 상세 검색의 적합성을 유지하면서도 검색 속도를 훨씬 더 향상시킬 수 있음이 실험을 통해 입증되었다.

## Abstract

This paper proposes the combined image retrieval system that gives the same relevance as exhaustive search method while its performance can be considerably improved. This system is combined with two different retrieval methods, and each gives the same results that full exhaustive search method does. Both of them are two-stage method. One uses condensation of feature vectors, and the other uses binary-tree clustering. These two methods extract the candidate images that always include correct answers at the first stage, and then filter out the incorrect images at the second stage. Inasmuch as these methods use equal algorithm, they can get the same result as full exhaustive search. The first method condenses the dimension of feature vectors, and it uses these condensed feature vectors to compute similarity of query and images in database. It can be found that there is an optimal condensation ratio which minimizes the overall retrieval time. The optimal ratio is applied to first stage of this method. Binary-tree clustering method, searching with recursive 2-means clustering, classifies each cluster dynamically with the same radius. For preserving relevance, its range of query has to be compensated at first stage. After candidate clusters were selected, final results are retrieved by computing similarities again at second stage. The proposed method is combined with above two methods. Because they are not dependent on each other, combined retrieval system can make a remarkable progress in performance.

**Keywords:** 이미지검색, 내용기반, 속도향상, 차원축약, 클러스터링

\* 정회원, LG전자 MC 사업본부 단말연구소

(Mobile Handset R&amp;D Center, Mobile Communication Company, LG Electronics Inc.)

\*\* 정회원, 제주대학교 컴퓨터교육과 (Department of Computer Education, Cheju National University)

\*\*\* 정회원, 한양대학교 정보통신대학 (Division of Information and Communications, Hanyang University)

접수일자: 2005년9월7일      수정완료일: 2006년1월9일

## I. 서 론

최근 영상 데이터의 양이 기하급수적으로 증가함에 따라 이를 효과적으로 색인하고 검색할 수 있는 방법이 요구되어 왔다. 기존의 키워드나 텍스트 기반의 검색 방법은 색인 구축과 객관성 유지의 어려움 때문에 내용 기반 검색 방법이 제안되었다. 내용기반 영상 검색은 영상 자체의 특징을 추출하여 이를 색인하고 이와 유사한 특징을 가지는 영상들을 검색하는 방법이다. 방대한 영상 데이터들을 단순히 파일로 저장하고 파일 이름과 키워드만으로 데이터를 관리하는 경우에는 내용기반 검색 기술의 필요가 없지만 이러한 데이터들을 정보로서 활용하고자 하는 경우에 내용기반 검색 기술의 중요성과 그 필요성은 절대적이다. 내용기반 검색의 핵심 기술은 크게 영상의 특징을 추출하는 기법, 고차원 특징 벡터를 이용한 검색에서 요구되는 효과적인 색인 기법, 그리고 두 멀티미디어 데이터 간의 유사도를 측정하는 기법 등으로 나눌 수가 있다. 그런데 영상을 나타내는 특징벡터가 일반적으로 고차원 벡터이기 때문에, 유사도 계산에 따르는 비용도 기존의 텍스트 기반의 검색에서 보다 훨씬 크게 된다. 또한 특징 벡터들 간의 유사도를 기반으로 검색이 이루어지므로 특징기술자(descriptor)의 종류와 차원 수가 시스템의 성능과 밀접한 관계를 가지고 있다.

내용기반 영상 검색에서는 고차원의 특징 벡터를 사용하면서도 효과적인 색인 작업을 하기 위하여 기존의 다차원 색인 기법(multidimensional indexing method)과 저차원 사상 기법(dimensionality reduction)을 같이 사용하고 있다. 다차원 특징 벡터를 이용한 기존의 정보 검색에서는 특징 벡터의 차원 수가 대략 10차원 이상이 되면, 선형 주사(linear scan)보다 성능이 나빠지기 시작한다. 이러한 이유로 고차원의 특징 벡터를 사용하는 경우에는 색인 기법을 사용하기 이전에 고차원의 벡터를 상대적으로 저차원의 벡터로 사상시키는 데에 관한 연구가 진행되었다. 저차원 사상 기법에 관련된 대표적인 방법으로는 부집합 선택(variable-subset selection), 다중차원 스케일링(MDS, multidimensional scaling), 기하해싱(geometric hashing)을 들 수가 있다. 부집합 선택 [8]은 원 특징 벡터의 기저 벡터들을 선형 변환에 의하여 사상시켜서 상대적으로 변화량(variance)이 적은 기저를 대상으로 차원을 줄인다. 이러한 선형 변환에 의한 방법은 차원을 줄일 경우, 근사치 오류를 최소로 할 수 있는 장점을 가지고 있다. KLT(Karhunen-Loève

Transform), SVD(Singular Value Decomposition), PCA(Principle Component Analysis) 등의 방법이 이에 속한다. 하지만 이러한 방법들은 데이터에 종속적이고 계산량이 많다는 단점을 가지고 있다. 다중차원 스케일링 [9][10]은 원 특징 벡터를 k-feature extraction function에 의해 비선형적인 방법으로 n차원에서 k( $k < n$ )차원으로 줄인다. SOM (self organized map)을 이용해 클러스터링을 한 후에 각 클러스터에 대해 선형 PCA를 적용시키는 비선형 PCA나 FastMap이 여기에 해당한다. 기하해싱 [11][12]은 영상의 정보를 기하학적인 관계에 의한 특징 점들로 표현을 하고, 이것을 동일한 크기의 해시 테이블에 저장을 한다. 차원 축소는 해싱 함수에 의해 이루어지는데 함수 설계가 복잡하고, 원 특징 벡터가 저차원일 경우에만 사용할 수 있다. 하지만 이러한 저차원 사상에 관한 연구는 상세 검색을 했을 경우의 적합성을 그대로 유지시키지 못하는 단점을 가지고 있다. 다차원의 특징 벡터로 이루어진 객체들의 검색에 있어서 적절한 기술자의 선택, 고차원 특징 벡터의 저차원으로서의 사상이 끝나게 되면 이제 적절한 색인 기법을 선택해야 한다. 일반적으로 사용되는 방법으로는 삼각부등식(triangle-inequality)의 성질을 이용하여 적절하지 않은 영상을 제거함으로써 전체 유사도 계산 횟수를 줄이는 방법, 벡터 공간 모델(vector space model), 그리고 척도 공간 모델(metric space model) [7] 등이 있다. 삼각부등식을 이용한 pruning algorithm [4][5][6]은 먼저 키를 선정한 뒤, 키와 데이터베이스 내의 다른 영상과의 유사도를 미리 계산하여 내림차순으로 정렬을 한다. 질의가 들어오게 되면 질의와 키와의 유사도를 계산하고, 이것을 이미 계산된 키와 데이터베이스 내 영상 사이의 유사도와 비교함으로써 불필요한 유사도 계산을 제거할 수가 있다. 많은 유사도 계산을 질의 영상이 들어오기 이전에 수행하고, 삼각부등식의 성질을 특징 벡터 사이의 거리에 적용하기 때문에 검색 속도를 향상시킬 수 있다. 이러한 방법은 적합성을 유지하면서 검색 속도를 향상시킬 수 있다는 장점이 있지만, 키의 개수와 종류에 상당히 종속적인 결과를 나타내고 최악의 경우 모든 유사도 계산을 다시 해야 할 수도 있다. 벡터 공간 모델은 유사성 검색을 위해서 가장 일반적으로 사용되는 모델이다. 벡터 공간 모델에서 객체는 벡터 공간상의 점으로 사상된다. 두 객체간의 유사성은 벡터 공간상의 두 점의 거리로 표현된다. 따라서 질의 객체와 유사한 객체를 검색하는 유사성 검색은 벡터 공간 모델에서 질의 점으로부터 가장 가까운 k개의 점을 검색하는 k-

최근접 검색(k-nearest neighbor search)이 된다. k-최근접 검색을 효율적으로 처리하기 위한 방법 중 분할 색인 구조(partitioning index structure)가 있다. 분할 색인 구조는 트리 구조로써 벡터 공간을 분할하고 각 분할 영역은 점들을 포함한다. 일반적으로 분할 영역을 MBR(Minimum Bounding Rectangle)이라고 한다. 이때 트리에서 같은 레벨의 서로 인접하는 MBR들은 상위 노드의 동일한 MBR에 포함된다. 그리고 트리의 루트 노드의 MBR은 벡터 공간의 전체 혹은 거의 대부분을 포함한다. 대표적인 분할 색인 구조에는 R-트리, k-d-트리가 있으며, 고차원 데이터를 위해 TV-트리, SS-트리, SR-트리 등이 제안되었다. 척도 공간 모델은 사용되는 유사도 측정 함수, 즉 metric이 계산하기에 너무 복잡하거나 데이터 집합에 따른 유사도가 제시될 때 사용된다. 일반적으로 대부분의 척도 공간 모델은 유사도 범위 내 검색(range query)에는 적절하지 못하지만, 최근접 검색(nearest neighbor search)에는 적절하게 설계되어 있다. 척도 공간 모델의 전통적인 방법으로는 vp-tree(vantage-point tree)가 있다. 각각의 노드는 연결되어 있지 않은 데이터베이스의 부집합을 가리키고 있고 두 개의 자식 노드를 가지고 있으며, 또한 vantage point와 연결되어 있다. 색인 된 영상들은 vantage point와의 거리를 기준으로 오름차순으로 정렬된다. 그리고 거리의 평균값이 계산되며 vantage point와의 거리가 평균거리 보다 작으면 왼쪽 자식노드로, 크면 오른쪽 자식노드로 구분하는 형태로 검색이 된다.

본 논문에서는 고차원 특징 벡터를 이용한 내용기반 검색에서의 효과적인 색인을 통한 검색 시스템의 성능 향상에 대한 연구 결과를 보인다. 본 논문에서는 전체 차원의 특징 벡터를 사용하여 순차적으로 검색했을 때의 결과와 동일한 적합성을 유지하면서 검색 시간을 현저히 줄일 수 있는 통합 검색 시스템을 제안한다. 통합 검색 방법은 차원 축약 검색 방법과 이진 트리 클러스터링 검색 방법의 두 가지가 통합되어 이루어져 있다. 차원 축약 2단계 검색 방법은 Cauchy-Schwartz 부등식의 성질을 이용하여 특징 벡터의 차원을 축약하여 적합성을 유지하면서 검색 속도를 향상시킬 수 있는 장점을 가지고 있다. 마찬가지로 이진 트리 클러스터링 2단계 검색 방법 역시 유사도 보정과정을 통하여 적합성을 유지하면서 검색 속도를 향상시킬 수 있다. 이러한 두 검색 방법이 서로 증속적이지 않다는 성질을 이용한 통합 검색 방법은 적합성을 유지하면서 각각의 방법보다 훨씬 뛰어난 검색 성능을 나타낸다는 것을 본 논문에서

는 입증한다. 각 검색 시스템의 성능을 비교하기 위하여 검색 시간의 거의 대부분을 차지하는 유사도 계산 복잡도를 모델링 하였다. 그리고 이는 특징 벡터의 차원에 무관하게 적용시킬 수 있는 장점을 가지고 있다.

## II. 특징 벡터의 차원 축약을 이용한 검색

내용기반 영상 검색에서는 일반적으로 100차원 이상의 고차원 특징 벡터를 사용하기 때문에 기존의 다차원 색인 기법을 적용할 수가 없다. 그리고 고차원의 특징 벡터를 저차원으로 사상시키기 위해 저차원 사상 기법을 사용한다면 기존의 적합성을 그대로 유지할 수가 없다. 그러므로 본 장에서는 기존의 적합성을 그대로 유지할 수 있는 고차원 특징 벡터의 차원 축약 기법에 대해서 설명한다<sup>[13]~[15]</sup>. 그리고 실험에 의해 검색 성능을 최대화시킬 수 있는 최적 차원 축약율이 존재한다는 것을 확인한다. 마지막 절에서는 특징 벡터의 차원 축약에 따른 유사도 분포가 베타분포 형태를 가진다는 사실에서, 통계적 모델링에 의해 임의차원의 특징 벡터의 최적 차원 축약율을 어느 정도 예측<sup>[16]</sup> 할 수 있다는 것을 보인다.

### 1. 각 Cauchy-Schwartz 부등식

원 특징 벡터의 차원 수를  $m$ , 전체 데이터베이스 내 영상 수를  $n$  이라고 하면, 데이터베이스 내의 영상에 대한 특징 벡터와 질의 영상에 대한 특징 벡터를 식 (1)과 같이 나타내고, 임의의  $m'$  차원으로 축약된 특징 벡터를 식 (2)와 같이 표시할 수 있다.

$$D^{(n)} = \{d_1, d_2, \dots, d_m\} \quad Q = \{q_1, q_2, \dots, q_m\} \quad (1)$$

$$D'^{(n)} = \{d'_1, d'_2, \dots, d'_{m'}\} \quad Q' = \{q'_1, q'_2, \dots, q'_{m'}\} \quad (2)$$

원 특징 벡터를 2배씩 축약시킬 경우, 축약된 특징 벡터 성분은 식 (3)에 의해 그 이전 단계의 특징 벡터의 각 성분보다 항상 크거나 같음을 알 수 있다.

$$d'_i = \sqrt{(d_{2i-1})^2 + (d_{2i})^2} \geq \max(d_{2i-1}, d_{2i}) \quad (3)$$

본 논문에서는 특징 벡터간의 유사도를 내적에 의한  $\cos\theta$  값으로 정의한다. 이러한 방법은  $L_2$  metric을 사용하면서도, 유사도 값을 0에서 1로 정규화 한다는 장점이 있다. 임의의 차원으로 축약된 특징 벡터에 의한 유사도 역시 원 특징 벡터를 사용해서 계산한 유사도 보

다 항상 크거나 같은 성질을 나타낸다. 이것은 식 (4)의 Cauchy- Schwartz 부등식에 의해 설명된다.

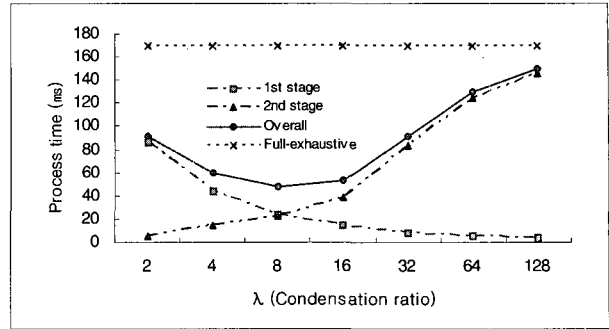
$$\sqrt{\{\sum d_i^2\} \cdot \{\sum q_i^2\}} \geq \sum (d_i \cdot q_i) \quad (4)$$

식 (4)의 좌변은 임의의 차원으로 축약된 특징 벡터로 계산한 유사도를 나타내고, 우변은 원 차원의 특징 벡터로 계산한 유사도를 나타낸다. 이로부터 임의의 차원으로 차원 축약된 특징 벡터로 계산한 유사도는 차원 축약율이 보다 큰 경우의 특징 벡터로 계산한 유사도보다 항상 크거나 같다는 것을 알 수 있다. 특징 벡터의 각 단계별 과정은 식 (5)에서 보다 상세하게 유도되어 있다. 식 (5)의 좌변은 원 차원의 특징 벡터로 계산한 유사도, 우변의 첫 항은 특징 벡터를 2배로 축약해서 계산한 유사도, 두 번째 항은 4배로 축약해서 계산한 유사도, 마지막 항은 n/2배로 축약해서 계산한 유사도이다. 우변의 각 항 역시 Cauchy-Schwartz 부등식을 만족하므로, 각 단계에서의 유사도는 그 이전 단계의 유사도보다 항상 크거나 같은 성질을 나타내며, 이러한 사실로 인하여 검색의 적합성이 보장된다.

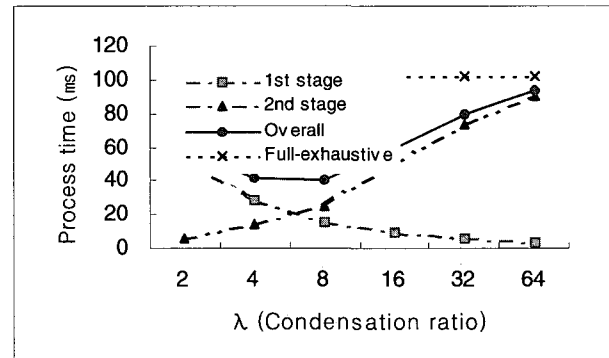
$$\begin{aligned} & (d_1q_1 + d_2q_2 + d_3q_3 + \dots + d_nq_n) \\ & \leq \sqrt{(d_1^2 + d_2^2)(q_1^2 + q_2^2)} + \sqrt{(d_3^2 + d_4^2)(q_3^2 + q_4^2)} + \dots + \sqrt{(d_{n-1}^2 + d_n^2)(q_{n-1}^2 + q_n^2)} \\ & \leq \sqrt{(d_1^2 + d_2^2 + d_3^2 + d_4^2)(q_1^2 + q_2^2 + q_3^2 + q_4^2)} + \dots + \sqrt{(d_{n-1}^2 + d_n^2 + d_{n-1}^2 + d_n^2)(q_{n-1}^2 + q_n^2 + q_{n-1}^2 + q_n^2)} \\ & \vdots \\ & \leq \sqrt{(d_1^2 + d_2^2 + d_3^2 + d_4^2 + \dots + d_n^2)(q_1^2 + q_2^2 + q_3^2 + q_4^2 + \dots + q_n^2)} \end{aligned} \quad (5)$$

2. 축약 2단계 검색방법

축약 2단계 검색 방법의 1단계 검색에서는 차원 축약된 특징 벡터를 이용하여 사용자가 제시한 유사도 기준 이상의 후보 영상들을 선정한다. 여기서 Cauchy-Schwartz 부등식의 특성에 의해 특징 벡터의 차원을 축약하기 때문에 상세 검색(exhaustive search)의 결과 영상들을 모두 포함한 후보 영상들이 선정된다. 2단계 검색에서는 1차 검색에서 선별된 후보 영상들을 대상으로 전체 차원의 특징 벡터를 이용하여 유사도를 재 계산하여 사용자 제시 유사도 기준 이상의 결과 영상들을 추출한다. 이러한 과정으로 인해 축약 2단계 검색 방법은 적합성을 그대로 유지하고 결과 순위에 영향을 미치지 않으면서도 검색 시간을 현저히 줄일 수 있다. 특징 벡터의 차원 축약율(condensation ratio) 를 식 (6)과 같이 정의할 때, 가 커질수록 1차 검색 시간이 줄어드는 반면, 1차 검색 결과인 후보 영상의 개수가 많아져 2차 검색



(a) 256차원



(b) 128차원

그림 1. λ값의 변화에 따른 응답시간  
Fig. 1. Response time with respect to different λ.

색 시간은 증가하게 된다.

$$\lambda = \frac{D_o \text{ (원 특징 벡터의 차원수)}}{D_c \text{ (축약 특징 벡터의 차원수)}} \quad (6)$$

반대로 λ값이 작아질수록 1차 검색 시간은 증가하는 반면, 후보 영상의 개수는 적어져 2차 검색 시간은 감소한다. 그림 1은 유사도 기준 0.8에서 원 특징 벡터가 256, 128 차원일 때, 각각 λ값의 변화에 따른 1차 검색 시간과 2차 검색 시간, 그리고 전체 검색 시간의 변화를 보여준다. 그림 3.5의 결과로부터, 축약 2단계 검색 방법에서 전체 검색 시간을 최소화 시키는 최적의 λ값이 존재함을 확인할 수 있다. 그리고 최적의 λ값을 λ<sub>opt</sub>라 하면, 특징 벡터의 차원이 256, 128 차원일 경우 λ<sub>opt</sub>는 8임을 알 수 있다.

III. 이진 트리 클러스터링을 이용한 검색

상세 검색의 적합성을 그대로 유지할 수 있는 다른 검색 방법으로 BTC(binary-tree clustering) [13][17]가 있다. 다차원 색인 구조에서 클러스터링은 많이 이용되

지만, 일반적으로 사용하는  $K$ -means 알고리즘을 이용한 클러스터링에서는 적절한  $K$ 를 찾기가 쉽지 않다. 따라서 본 논문에서는 데이터베이스에 따라 클러스터의 개수  $K$ 가 동적으로 결정될 수 있고, 또한 데이터베이스 내에 새로운 영상이 삽입되거나 삭제될 경우 색인 구조의 변경이 용이한 BTC를 이용한 2단계 검색 방법을 적용한다.

1.  $K$ -means 알고리즘

$K$ -means 알고리즘은 클러스터 검출에서 가장 보편적으로 이용되는 방법으로 사전에 결정된 클러스터의 수  $K$ 에 기초하여 전체 데이터를 상대적으로 유사한  $K$ 개의 클러스터로 구분하는 방법이다<sup>[18]</sup>. 그림 2의 (a)에서 데이터들은 두 개의 클러스터들로 수렴하며 새로운 위치는 (b)에서처럼 각각의 클러스터에 속하는 데이터들의 평균 위치가 된다.

$K$ -means 알고리즘은 비감독 신경망 클러스터링 알고리즘인 CL(competitive learning)<sup>[19]</sup>과 RPCL(rival penalized competitive learning)<sup>[20]</sup> 및 VP-tree<sup>[21]</sup>에 비해 계산 시간이 약간 더 걸리는 단점이 있지만, 데이터들을 분류하는데 있어 좀 더 정확한 결과를 보장해 준다. 여기서는  $K$ -means 알고리즘을 사용하여 영상의 특징 벡터가 저장되어 있는 데이터베이스 내의 데이터들을 유사한 그룹으로 분류하였다.

2. 이진 트리 클러스터링

$K$ -means 알고리즘을 비롯하여 다른 클러스터링 알고리즘은 초기의 입력 벡터의 개수에 따라 그룹화되는 데이터들의 개수가 정해진다는 단점이 있다. 그러나 영상 데이터베이스 안의 특징 벡터들을 유사한 벡터들끼리 그룹화 할 때에는 몇 개의 그룹으로 그룹화 해야 할지 사전에 정할 수 없는 경우가 대부분이다. 때문에 그룹의 개수를 상황에 따라서 동적으로 결정하는 알고리즘이 필요하다. BTC는 그림 3과 같이 그룹의 개수를

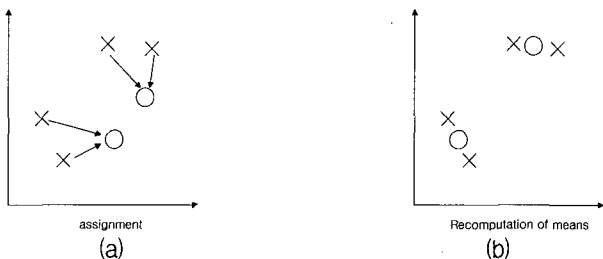


그림 2.  $K$ -means 알고리즘  
Fig. 2.  $K$ -means algorithm.

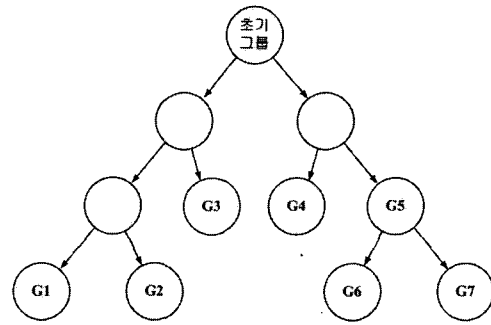


그림 3. 이진 트리 클러스터링  
Fig. 3. Binary tree clustering.

트리 구조 형태로 두 개씩 늘려가면서 적절한 수의 그룹 개수를 결정하는 방법이다.

먼저 전체 데이터베이스를 루트노드라 가정하고, 여기서 임의의 seed 값  $s_1, s_2$ 를 선정하여 2-means 알고리즘을 적용하여 수렴시킨다. 이 때, 식 (7)의 각 자식 노드의 클러스터 반경  $r_{c_i}$ 이 사용자가 지정한 클러스터링 유사도  $\mu_{cb}$ 보다 크다면 각 자식 노드에서 다시 2-means 알고리즘을 적용하여 클러스터링을 수행한다. 이 과정을 모든 잎 노드에서 만족할 때까지 반복하면 이진 트리 클러스터링이 완료된다. 식 (7)에서  $o_{c_i}$ 는  $i$  번째 클러스터  $C_i$ 의 중심을,  $x_{c_i}$ 는  $i$  번째 클러스터  $C_i$ 에 속해 있는 원소  $x$ 를 나타낸다. 그러므로  $r$ 은 클러스터의 중심과 그 클러스터 내의 원소 중 가장 멀리 떨어져 있는 원소와의 유클리드 거리를 의미한다. 위의 이진 트리 클러스터링 과정의 알고리즘을 그림 4에 나타내었다.

$$\mu_{cb} = \max \left( \text{dist}_{L_2} \left( o_{c_i}, x_{c_i} \right) \right) \tag{7}$$

그림 4의 알고리즘은 실제 부모노드가 2-means 클러스터링 되어 두 자식노드로 분할되는 하나의 함수가 내부에서 재귀적으로 호출되는 형태로 이루어져 있다.

BTC Algorithm

1. 부모 노드에서 임의의 두 seed  $s_1, s_2$  선정
2.  $s_1, s_2$ 를 초기값으로 하여 2-means 클러스터링 (수렴시까지)
3. if ( 왼쪽 자식 노드의  $r$  ) >  $\mu_{cb}$   
 위의 1, 2의 과정 반복  
 else if ( 오른쪽 자식 노드의  $r$  ) >  $\mu_{cb}$   
 위의 1, 2의 과정 반복  
 else  
 클러스터링 완료

그림 4. 이진 트리 클러스터링 알고리즘  
Fig. 4. Binary tree clustering algorithm.

### 3. 이진 트리 클러스터링을 이용한 2단계 검색 방법

BTC를 이용한 2단계 검색 방법은 사용자가 입력하는 질의 영상의 특징 벡터와 각 클러스터의 중심 벡터 간의 1단계 검색을 통해 2단계 상세 검색을 수행할 검색 영역을 최소한으로 선정함으로써 유사도 비교 시간을 단축시키는데 목적이 있다. 각 클러스터의 대표 벡터는 클러스터 내에 소속된 특징 벡터들의 평균값으로 정의한다. 1단계 검색에서는 각 클러스터들을 대상으로 질의에 대한 사용자가 제시한 유사도 기준을 만족하는 후보 클러스터를 선정하고, 2단계 상세검색에서는 후보 클러스터 내의 영상만을 대상으로 유사도 비교를 통해 최종 검색 결과를 얻는다. 이런 과정을 통하여 유사도 비교대상이 되는 영상의 수를 줄일 수 있기 때문에 전체 검색 속도를 향상시킬 수가 있다. 즉, 특징 벡터들을 BTC 알고리즘을 이용하여 그룹화시켜 저장하면 모든 특징 벡터들을 비교하지 않더라도 전체를 검색한 것과 동일한 결과를 얻을 수 있다.

### 4. 유사도 보정

질의가 들어왔을 때, 사용자가 제시한 유사도 기준을 1단계 검색과 2단계 검색에서 동일하게 적용시키면 상세 검색 시의 적합성을 유지하지 못하는 경우가 발생한다.  $\mu_{r,q}$ 를 질의와 각 클러스터의 대표 벡터와의  $L_2$  distance,  $\mu$ 를 사용자가 제시한 유사도 기준이라고 하면 식 (8)의 관계에 의해 검색 결과를 얻어낸다.

$$\mu_{r,q} \leq \mu \quad (8)$$

이 때 대표 벡터 가운데 그림 5에서의 G3와 같이 클러스터와 질의와의 유사도가 사용자가 제시한 기준에는 만족하지 못하지만, 그 클러스터에 속해있는 벡터들 중에 질의와의 유사도 기준에 만족하는 경우가 발생할 수 있다. 이런 경우 최종 검색 결과에서 제외되며, 상세 검색과 비교하여 정확도는 100%를 유지하나 재현율은 떨어지게 된다. 이러한 문제점을 해결하기 위하여 식 (9)와 같이 사용자가 제시한 유사도 기준을 클러스터링 유사도 만큼 확장하도록 보정해 주며, 이와 같은 방법을 그룹 경계 보정이라 정의한다.

$$\mu_c = \mu + \mu_{cb} \quad (9)$$

여기서,  $\mu_c$ 는 보정된 유사도이다. 이 유사도를 적용하면 클러스터링을 이용한 검색한 결과와 전체를 검색

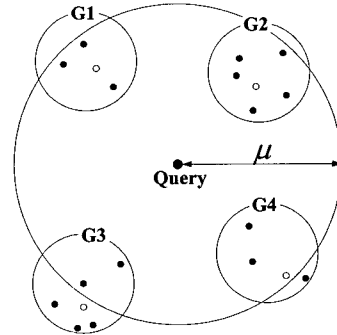


그림 5. 후보 그룹 선정 오류

Fig. 5. An error of the selection of the candidate group.

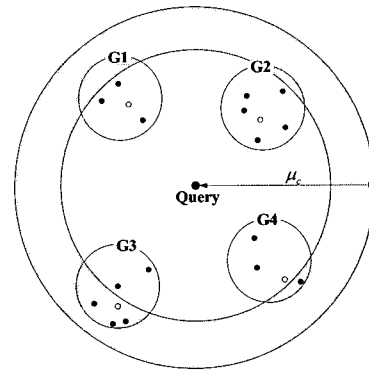


그림 6. 그룹 경계 보정

Fig. 6. Compensation of the group boundary

한 결과에 대해 정확도, 재현율, 순위 등에서 모두 동일한 결과를 얻을 수 있다. 그러므로 적합성이 유지되는 검색 결과를 얻기 위해 각 단계에서 식 (8)의 유사도 기준을 동일하게 적용시키지 않고 식 (10)처럼 각 단계의 유사도 기준을 따로 적용시킨다.

$$\begin{cases} \mu_{r,q} \leq \mu_c & : 1 \text{ 단계} \\ \mu_{r,q} \leq \mu & : 2 \text{ 단계} \end{cases} \quad (10)$$

그림 6은 식 (10)을 사용하여 그룹 경계 보정을 하였을 경우 검색영역이 점선으로 나타나는 바와 같이 확장되어 문제가 되었던 G3를 후보 그룹으로 선정하고 있음을 보여주고 있다.

## IV. 통합 검색시스템

III장에서 언급한 축약 2단계 검색 방법과 BTC를 이용한 2단계 검색 방법은 모두 1단계 검색에서 후보 영상들을 선정하고, 2단계의 상세 검색에서는 후보 영상을 대상으로 재 검색 과정을 거친다. 그러므로 두 방법 모두 상세 검색을 했을 때의 적합성을 그대로 지닌다. 또한 이 두 검색 방법은 서로 과정이 종속적이지 않고 독립적이기 때문에 두 방법을 동시에 적용시킨다면, 각각

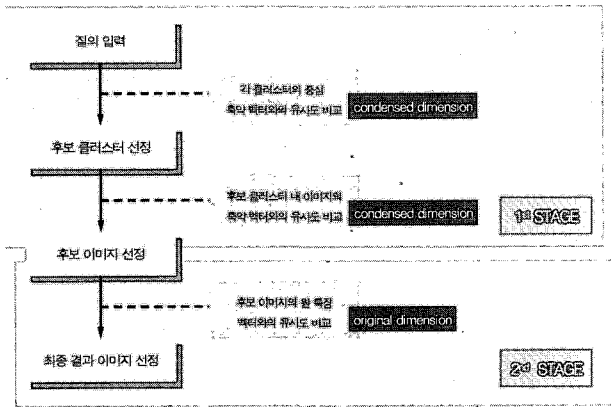


그림 7. 통합 검색의 시스템도  
Fig. 7. Schematic diagram of an integration retrieval.

의 검색 방법보다 검색 성능이 더 향상될 것이다. 이에 본 논문에서는 두 검색 방법의 알고리즘의 혼합을 통한 통합 검색 시스템을 제안하고, 통합 검색 시스템의 성능이 각각의 두 검색 방법의 성능보다 훨씬 더 우수한 것을 보인다.

1. 통합 검색 시스템의 구성

통합 2단계 검색의 경우, 전체 시스템의 구성을 그림 7에 보였다. 여기서 앞에서 언급한 축약 2단계 검색 방법과 BTC를 이용한 2단계 검색 방법이 통합되어 있다. 데이터베이스 내의 영상들은 원 차원의 특징 벡터와 축약 벡터를 동시에 가지고 있으며, 이미 특정한 유사도 ( $\mu_{cb}$ ) 보다 큰 영상들끼리 서로 클러스터링 되어 있다. 질의 영상이 들어오게 되면 먼저 질의의 원 차원 특징 벡터를 추출하고, 이것을 바탕으로 식 (3)에 의해 축약 벡터를 생성하게 된다. 1단계 검색에서는 질의 영상의 축약 벡터와 각 클러스터의 축약 중심 벡터 사이의 유사도를 계산하여 후보 클러스터를 선정한다. 그리고 질의 영상의 축약 벡터와 후보로 선정된 클러스터 내에 포함되어 있는 영상들의 축약 벡터 사이의 유사도를 계산하여 후보 영상들을 선정하게 된다. 2단계 검색에서는 질의 영상의 원 특징 벡터와 후보 영상들의 원 특징 벡터를 사용하여 최종 결과 영상을 선정하게 된다.

2. 데이터베이스 구축 단계

질의가 들어오기 이전 데이터베이스 구축 단계에서 가장 먼저 모든 데이터베이스 내의 영상을 대상으로 BTC가 이루어진다. 재귀적인 과정에 의해 모든 클러스터가  $\mu_{cb}$ 의 조건을 만족하게 되면 첫 번째 과정은 완료된다. 두 번째로 각 클러스터 내의 영상에 대해 최적

차원 축약을  $\lambda_{opt}$ 를 적용하여 축약한 여벌의 특징 벡터를 계산하여 각각의 클러스터 별로 저장한다. 이 단계가 완료되면, 하나의 영상은 자신이 속한 클러스터 내에서 원 차원의 특징 벡터와 차원 축약된 특징 벡터를 모두 지니고 있게 된다. 이것으로 질의가 들어오기 전까지의 모든 과정이 완료된다. 데이터베이스 크기가 클수록 고 차원의 특징 벡터를 사용할수록 이 과정의 시간은 오래 걸리지만, 질의가 들어와서 검색 결과를 얻을 수 있기까지 즉, 실제 검색 시스템의 사용에 있어서의 효율은 아주 높아진다.

3. 1단계 검색 과정

질의가 들어오게 되면, 먼저 질의 영상에 대해 해당 기술자로 전체 차원의 특징 벡터를 추출하게 된다. 그리고 전체 차원의 특징 벡터를 바탕으로  $\lambda_{opt}$ 를 적용하여 축약 특징 벡터를 추출한다. 그리고 질의 영상의 축약된 특징 벡터와 각 클러스터 간의 대표 벡터의 축약 특징 벡터와의 유사도를 계산한다. 이 유사도가 사용자가 제시한 유사도 기준을 만족하면 해당 클러스터를 후보 클러스터로 결정한다. 축약된 특징 벡터를 이용하여 선정된 후보 클러스터 내의 모든 영상들을 대상으로 질의와의 유사도를 재 계산한다. 이 유사도가 사용자가 제시한 유사도 기준을 만족하면 최종 후보 영상들이 결정된다. 이 과정이 모든 클러스터에 대해 적용되어 2단계 상세 검색을 하기 위한 모든 후보 영상들이 선정된다.

4. 2단계 검색 과정

통합 검색 시스템의 2단계 검색 과정은 차원 축약 2단계 검색 방법과 BTC를 이용한 2단계 검색 방법의 두 번째 단계의 검색 과정과 동일하다. 1단계 검색 과정에서 얻은 결과가 모두 후보 영상들이 된다. 이 후보 영상들을 대상으로 원 차원의 특징 벡터를 사용하여 사용자 제시 유사도 기준으로 재 검색을 실시한다. 최종 검색 결과로 나오는 영상들은 모두 상세 검색의 결과 영상들과 일치하는 적합성을 유지하고 있고, 각 검색 방법의 성능보다 훨씬 더 우수한 것을 알 수 있다.

5. 통합 검색에서의 유사도 보정

차원 축약된 특징 벡터를 이용하여 동일한 유사도 기준으로 검색을 한다는 것은 사용자 제시 유사도 기준을 좀 더 낮추어 검색한다는 것과 동일한 의미이다. 하지만 이 두 관계에 대한 정량적인 분석은 쉽지가 않다. 통합

검색 방법에서는 차원 축약된 특징 벡터를 사용하여 후보영상을 추출해내기 때문에, 실제로 유사도 기준이 사용자 기준의 유사도보다 더 낮아져서 어느 정도의 유사도 보정 역할을 한다는 사실을 알 수가 있다. 거기에 이진 트리 클러스터링에서의 유사도 보정 과정이 동일하게 적용되면, 유사도 보정이 필요 이상 되어 후보 영상이 필요 이상으로 선정되게 된다. 실제로 통합 검색에서 유사도 보정을 하지 않았을 경우, 약 91~93%의 적합성이 유지되지만 성능은 유사도 보정을 한 경우보다 훨씬 더 향상되었다. 따라서 본 논문에서는 별도의 유사도 보정 과정을 거치지 않은 검색 성능과 유사도 과정을 거친 경우의 검색 성능 두 가지를 모두 제시한다.

## V. 실험 결과

### 1. 실험 환경

본 실험은 Pentium III 866MHz, 256 RAM 환경에서 이루어졌고, 영상데이터는 MPEG-7의 CCD(common color dataset) 5466장과 CCQ(common color query) 50장으로 이루어졌다. Descriptor로는 256, 128차원의 CSD(color structure descriptor)를 사용하였고, 복잡도는 특징 벡터의 차원에 무관하게 정의된 유사도 계산횟수  $N$ 으로 정의하였다.

### 2. 복잡도 모델링

검색 시스템의 성능을 평가하기 위한 방법으로 사용자에게 의해 질의가 입력된 이후 최종 결과를 얻을 때까지의 응답 시간을 측정할 수가 있다. 하지만 이러한 방법은 실험하는 환경에 민감하여 성능을 평가하기 위한 수단으로는 적절하지 않다. 전체 응답시간에 영향을 미치는 요소로는 질의 영상에 대한 특징 벡터를 추출하는 데 걸리는 시간, 데이터베이스 내의 영상들과 유사도 비교를 위한 유사도 계산 시간, 그리고 정렬하는 데 걸리는 시간으로 크게 나눌 수 있다. 일반적으로 내용기반 영상 검색에서는 고차원의 특징 벡터를 사용하므로 결국 질의와 데이터베이스 내의 유사도 계산에 가장 많은 시간이 걸린다. 이러한 이유로 본 논문에서는 검색 시스템의 성능을 비교 평가하기 위한 기준으로 유사도 계산 횟수를 사용한다 [5]. 전체 차원의 특징 벡터를 사용하여 유사도 계산 1회에 걸리는 복잡도를  $N$  이라고 정의하고,  $\lambda$  값이  $2^k$  형태로 증가하게 되면 각 경우 복잡도는  $N/2^k$ 가 된다. 이러한 방법을 사용하게 되면 차원 변화에 상관없이 복잡도를 측정할 수가 있고, 각 검색

방법의 성능을 비교 평가할 수 있는 기준이 된다.

### 3. 실험 결과

5.2절에서 정의한 복잡도를 이용하여 각 검색 방법의 성능을 비교한다. 가장 먼저 전체 차원의 특징 벡터를 순차적으로 적용한 상세 검색 방법을 나머지 검색 방법의 성능 향상 정도를 평가하기 위한 기준으로 제시한다. 삼각부등식을 이용한 검색은 키 선정에 있어서 random 방법과, BTC에서 사용된 각 클러스터의 중심을 키로 선택하였을 경우의 검색 성능을 보인다. 특징 벡터의 차원 축약을 이용한 2단계 검색에서는 최적 차원 축약을  $\lambda_{opt}$ 를 적용하였을 경우의 검색 성능을 보인다. BTC를 이용한 2단계 검색에서는 클러스터 유사도 기준  $\mu_{cb}$ 가 변할 경우, 최종적으로 나타나는 클러스터 개수에 대해 언급하고, 일반적으로 향상되는 검색 성능의 정도를 보인다. 그리고 통합 검색 시스템에서는 4.5절에서 언급한 것처럼 유사도 보정을 하지 않았을 경우와 유사도 보정을 했을 경우에 최종 결과와 성능 향상 정도를 함께 비교 제시한다. 마지막으로 데이터베이스의 크기가 변할 경우 통합 검색의 성능의 변화를 제시하여, BTC 알고리즘이 대용량 데이터베이스에서 좋은 성능을 나타내는 것을 보인다.

#### 가. 상세 검색 방법의 성능

전체 차원의 특징 벡터로 순차적인 상세 검색을 한 경우의 성능은 질의와 데이터베이스 내의 영상 사이의 일대일 유사도 계산이 이루어지므로, 데이터베이스의 크기  $\times N$  이 성능이 되고, 본 논문에서는 CCD를 사용하였기 때문에 차원에 관계없이 성능은  $5466N$  이 된다. 상세 검색의 성능은 다른 검색 방법들과의 비교를 위한 기준으로 제시하였다.

#### 나. 삼각부등식을 이용한 검색 방법의 성능

삼각부등식을 이용한 검색은 random 키를 선택하였을 경우와 클러스터의 중심을 키로 선택하였을 경우에 대하여 유사도 기준 0.9~0.7 일 때 비교 실험을 진행하였다. random 키를 사용한 경우, 각각 원 특징 벡터의 차원 수가 256차원 일 경우와 128 차원일 경우가 그림 8과 그림 9에 제시하였다. random 키를 사용했을 경우에는 키의 선정에 따라 같은 수의 키를 사용하더라도 성능이 조금씩 다르게 나오므로, 각각 5번씩 실행한 결과의 평균값을 적용하였다.

그럼에도 불구하고 그래프에서 튀어 나오는 부분은



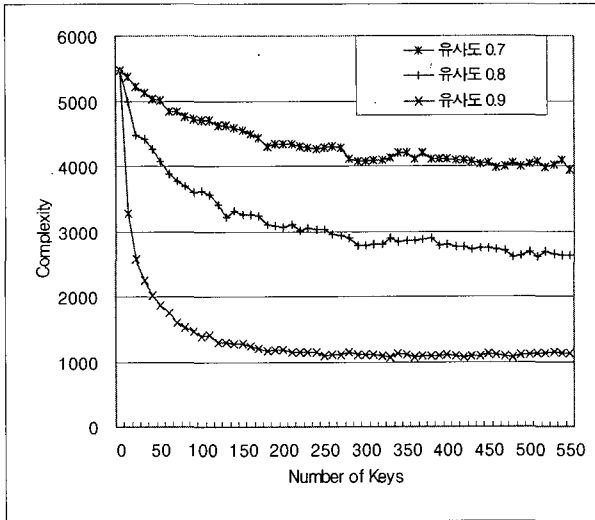


그림 8. random 키를 사용했을 때, 키의 수에 따른 복잡도의 변화 (256차원)  
 Fig. 8. Complexity with respect to the number of key (Use random key, 256).

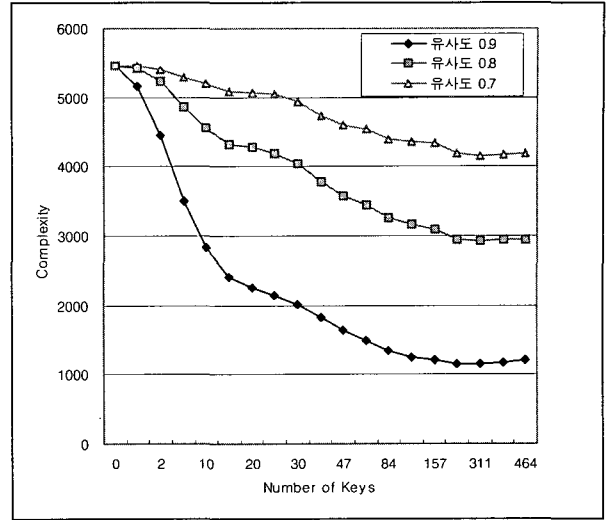


그림 10. BTC 클러스터의 중심을 키로 사용했을 때, 키의 수에 따른 복잡도의 변화 (256차원)  
 Fig. 10. Complexity with respect to the number of key (Use center key of cluster, 256).

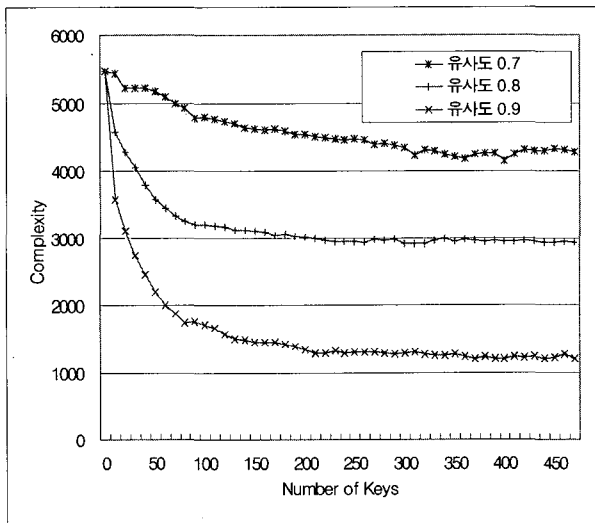


그림 9. random 키로 사용했을 때, 키의 수에 따른 복잡도의 변화 (128차원)  
 Fig. 9. Complexity with respect to the number of key (Use random key, 128).

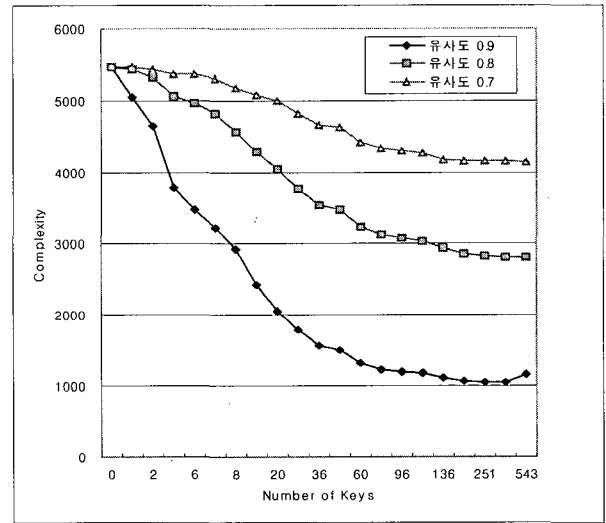


그림 11. BTC 클러스터의 중심을 키로 사용했을 때, 키의 수에 따른 복잡도의 변화 (128 차원)  
 Fig. 11. Complexity with respect to the number of key (Use center key of cluster, 128).

random 키 방법이 어느 정도의 성능은 보장해 주지만, 키에 따라 성능이 불규칙하게 저하될 수도 있다는 것을 의미한다. 삼각부등식을 사용한 검색에서는 키의 선정도 중요하지만, 얼마나 많은 수의 키를 사용하는 것이 적절한가도 중요한 문제이다. 그림 8과 그림 9에서 키의 개수가 많으면 많을수록 검색 성능은 향상되지만, 키의 개수가 어느 정도 이상 많아지면 성능이 포화된다는 것을 알 수가 있다. 256 차원에서는 200~250개 사이에서 성능이 포화되고, 128 차원에서도 200~250개 사이에서 포화된다. 실험 결과에서 알 수 있듯이 키가 일정 개수

이상 많아져도 성능은 비슷하지만, 결국 데이터베이스 구축단계에서 키와 데이터베이스 내 영상 사이의 distance를 전부 계산하여 저장하고 있어야 하므로 데이터베이스 관리상 많은 비용을 들여야 한다.

그림 10과 그림 11은 삼각부등식 검색 방법과 3.3절의 BTC 검색 방법을 혼합한 결과이다. BTC 클러스터의 중심을 삼각부등식에 사용되는 키로 정의하였다. BTC는 이진 트리 형태로 단계적으로 조건에 의하여 클러스터링 되기 때문에 각 단계의 클러스터 중심을 키로 사용하고 이에 따른 검색 성능을 보여준다. 그림 10은

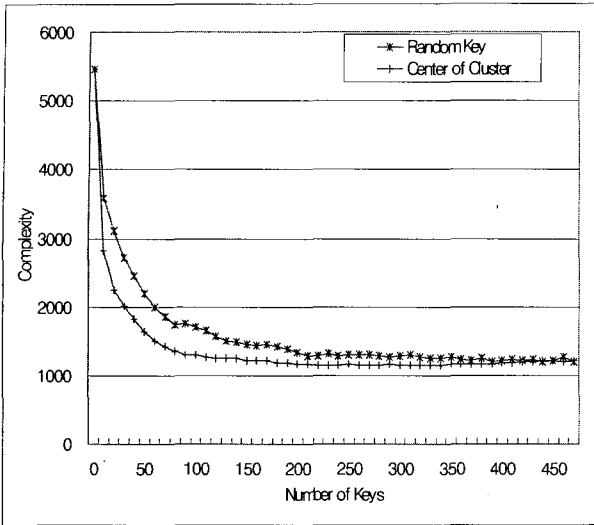


그림 12. 키 선정 방법에 따른 삼각부등식 검색의 성능 비교 (유사도 0.9)  
 Fig. 12. Performance comparison of triangle inequality with respect to the key selection method (Similarity 0.9).

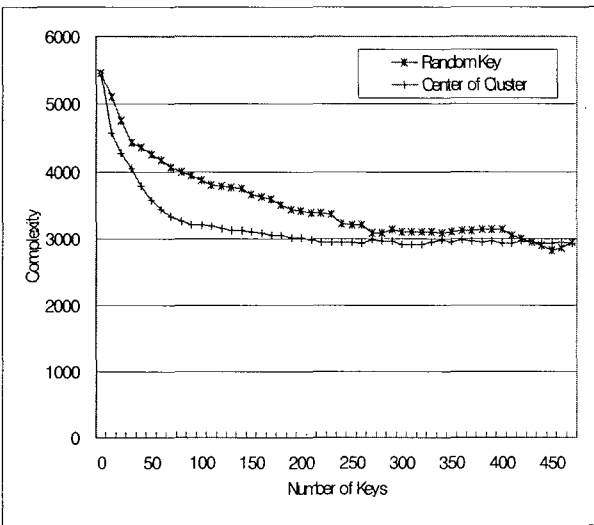


그림 13. 키 선정 방법에 따른 삼각부등식 검색의 성능 비교 (유사도 0.8)  
 Fig. 13. Performance comparison of triangle inequality with respect to the key selection method (Similarity 0.8).

256 차원에서, 그림 11은 128 차원에서의 복잡도를 나타내고 유사도 기준 0.9~0.7에서의 복잡도를 각각 나타내었다. 데이터베이스 내의 영상들을 k-means 알고리즘을 이용하여 클러스터링을 하고 이의 중심을 키로 사용하였으므로, 같은 수의 키를 사용하였을 때 random 방법보다 성능이 훨씬 우수하다.

그림 12과 그림 13은 특징 벡터가 256 차원일 경우, random 방법과 BTC 클러스터의 중심을 키로 사용한 방법의 성능을 비교하고 있다. random 방법은 키의 개

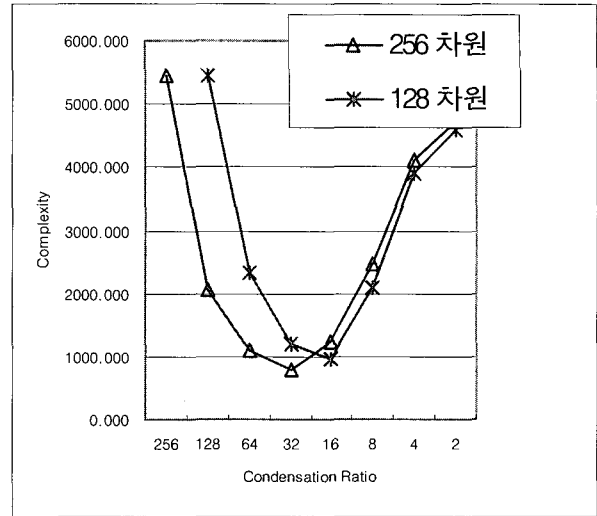


그림 14. 차원에 따른 축약 2단계 검색 방법의 성능  
 Fig. 14. Performance of the condensation two-stage retrieval method with respect to the dimension.

수가 200~250 사이에서 성능이 포화되지만, BTC 클러스터 방법은 100~150 사이에서 성능이 포화된다. 이것은 키를 잘 선정하게 되면 보다 적은 수의 키를 가지고도 더 나은 성능을 나타낼 수 있다는 사실을 말해준다. 그리고 삼각부등식 검색은 유사도 기준 별 정답을 100% 만족하는 적합성을 보장한다. 하지만 실험 결과에서 나타나는 것처럼 포화되었을 경우가 삼각부등식 검색의 한계 성능이 된다.

다. 차원 축약 2단계 검색 방법의 성능

그림 14은 차원 축약 검색 방법을 이용하였을 경우의 검색 성능을 나타낸다. 각 차원별로 최소 값을 나타내는 곳이  $\lambda_{opt}$ 가 되는 지점이고, 통합 검색에서는  $\lambda_{opt}$  일 경우를 적용한다. 차원 축약에 의한 검색 방법도 마찬가지로 상세 검색과 동일한 적합성을 보장한다.

라. 이진 트리 클러스터링을 이용한 2단계 검색 방법의 성능

그림 15와 그림 16은 클러스터 유사도 기준 0.7에서 이진 트리 클러스터링을 이용한 검색의 성능을 나타낸다. 각각은 원 특징 벡터가 256 차원, 128 차원일 경우를 나타내며, 상세 검색과의 성능을 비교해서 제시하고 있다. 이진 트리 클러스터링의 경우, 각 단계에서 키를 random하게 선택하기 때문에 같은 데이터베이스를 클러스터링 할 경우에도 매번 최종 클러스터 개수가 조금씩 다르게 나온다. 같은 조건에서 약간의 성능 오차가

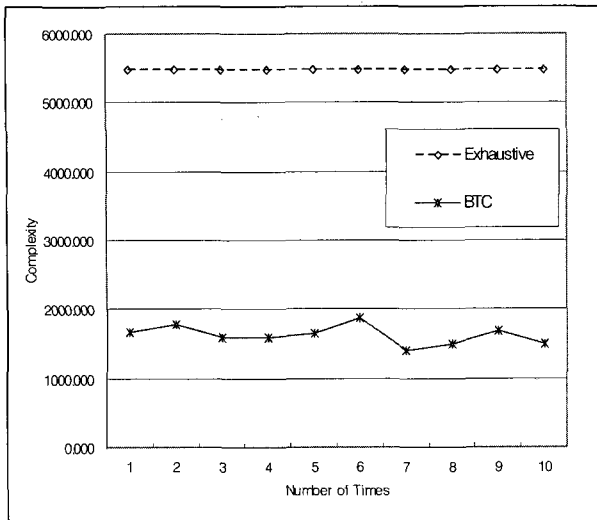


그림 15. BTC를 이용한 검색의 성능 (256 차원)  
Fig. 15. Retrieval performance using BTC (256).

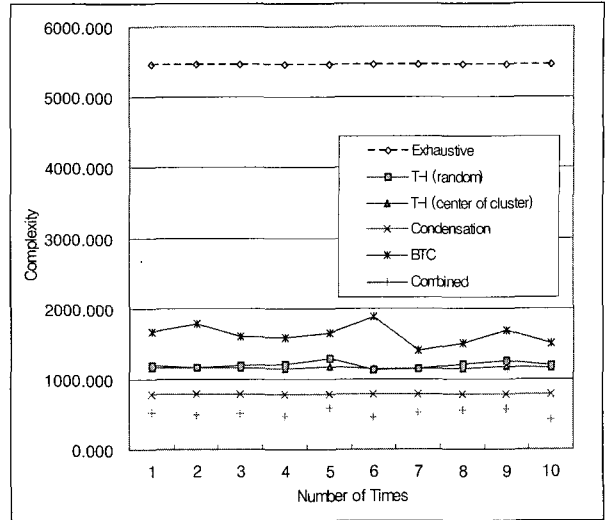


그림 17. 통합 검색의 성능 비교 (256 차원)  
Fig. 17. Performance comparison of an integration retrieval (256).

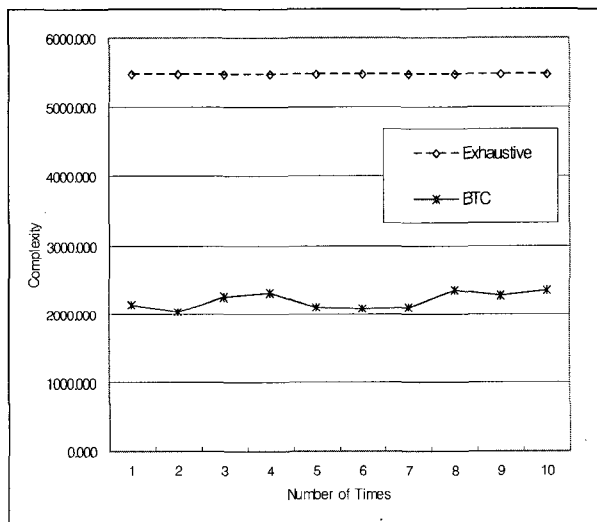


그림 16. BTC를 이용한 검색의 성능 (128 차원)  
Fig. 16. Retrieval performance using BTC (128)

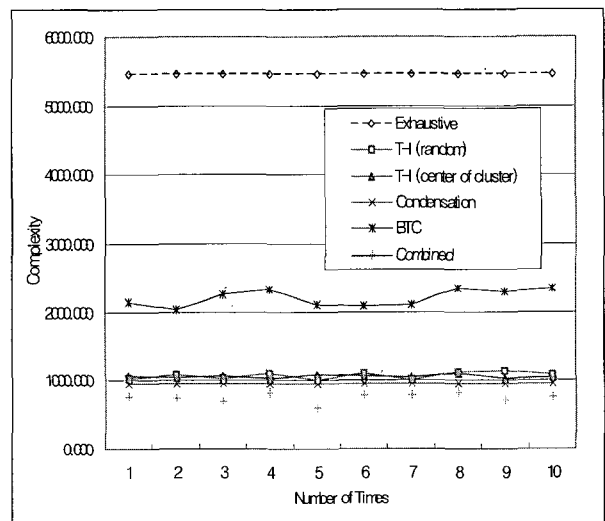


그림 18. 통합 검색의 성능 비교 (128 차원)  
Fig. 18. Performance comparison of an integration retrieval (128).

발생한다는 문제점이 있긴 하지만, 각 클러스터 반경을 동일하게 제한함으로써 인하여 데이터베이스에 따라 동적으로 분할될 수 있고 영상의 삽입, 삭제에도 편리한 동적 데이터베이스 검색 시스템이 가능하다는 장점이 있다.

그리고 이진 트리 클러스터링을 이용하게 되면, 데이터베이스의 크기가 커질수록 성능이 향상된다는 것을 알 수가 있다. 복잡도가 데이터베이스의 크기가 커짐에 따라 선형적으로 증가하는 것이 아니라 로그 함수의 형태로 증가하는 사실을 알 수가 있는데, 이 부분에 대한 자세한 설명은 그림 21과 그림 22에서 다시 언급한다.

마. 통합 검색 방법의 성능

그림 17과 그림 18은 본 논문에서 제안한 통합 검색

방법에 대한 성능을 보여준다. 상세 검색은 차원과 무관하게 복잡도는 데이터베이스의 크기가 된다. random 키를 이용한 삼각부등식 검색 방법은 검색 성능이 포화되었을 경우 10번 반복하여 나온 결과이다. BTC의 클러스터 키를 이용한 방법은 클러스터 유사도 기준 0.7로 10회 클러스터링을 시켰을 때, 각각의 클러스터의 중심을 키로 선정 시 검색 성능을 나타낸다. 여기서도 마찬가지로 키의 개수는 포화된 경우를 적용하였다. 차원 축약 검색 방법은 256, 128 각 차원에서  $\lambda_{opt}$ 인 경우의 2단계 검색 결과를 나타낸다. BTC 검색 방법도 클러스터링 유사도 기준 0.7로 반복해서 10회 클러스터링 시켰

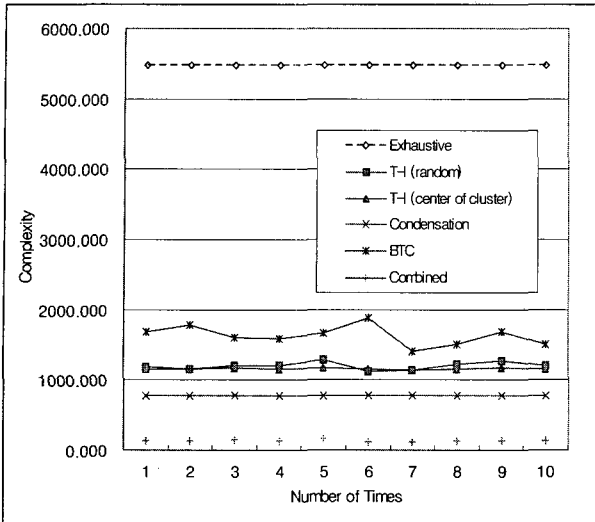


그림 19. 유사도 보정을 하지 않았을 경우의 통합 검색 성능 (256 차원)  
 Fig. 19. Integration retrieval performance (Not using similarity compensation, 256).

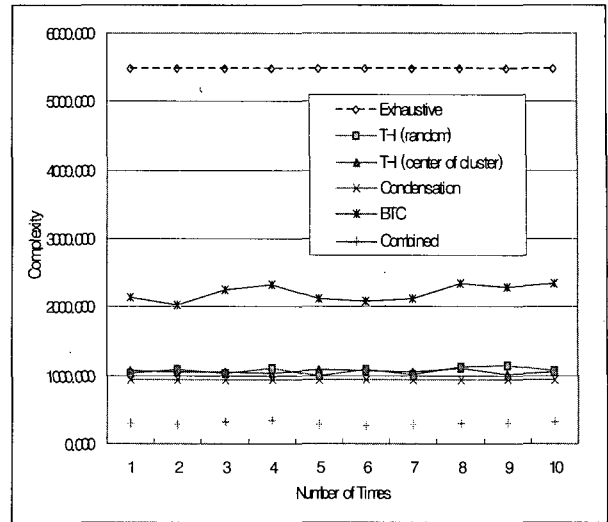


그림 20. 유사도 보정을 하지 않았을 경우의 통합 검색 성능 (128 차원)  
 Fig. 20. Integration retrieval performance (Not using similarity compensation, 128).

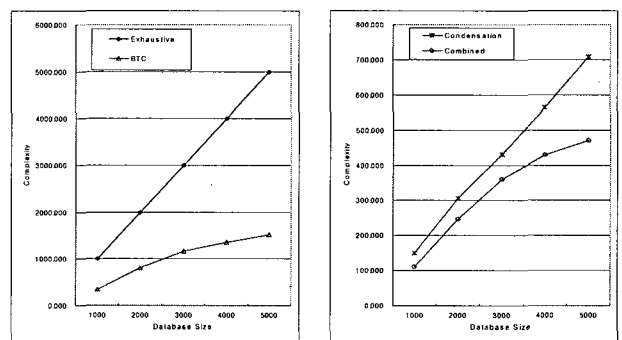
을 때, 2단계 검색 결과이다. 마지막으로 통합 검색 방법의 성능은 클러스터링 후 유사도 보정 과정을 거친 다음, 축약 검색의 방법이 적용된 결과이다.

실험 결과에서 살펴보면 유사도 보정을 했을 경우, 본 논문에서 제안한 통합 검색 방법의 성능이 상세 검색의 성능보다 256 차원에서 10.424배, 128 차원에서 7.217배 향상된 것을 확인할 수가 있다. 이 경우 상세 검색의 적합성을 100% 유지한다.

그림 19와 그림 20은 제안한 통합 검색 방법에서 유사도 보정 과정을 거치지 않았을 경우의 결과를 보여준다. 4.5절에서 언급한 것처럼 통합 검색 방법의 특징 벡터의 차원 축약 단계에서 어느 정도 유사도 보정 과정이 일어난다. 하지만, 이러한 방법으로는 적합성을 완전히 보장할 수는 없다. 유사도 보정을 하지 않았을 경우 256 차원과 128 차원에서 각각 93%, 91%의 적합성을 보장하였다. 하지만 이 경우의 성능은 각각 39.954배, 18.501 배 향상되었다. 이것은 통합 검색에서의 유사도 보정 과정이 실제로 많은 redundancy를 포함하고 있다는 사실을 말해준다. 하지만 통합 검색에서의 적절한 유사도 보정은 이 부분에 대한 정량적인 분석을 하기가 어렵기 때문에 쉽지 않다. 불필요한 유사도 계산을 최소화시킬 수 있는 부분의 연구가 추후 진행되어야 할 것이다.

바. 데이터베이스 크기 변화에 따른 성능

그림 21은 데이터베이스 크기에 따른 복잡도 변화를 나타낸다. 상세 검색과 차원 축약 검색 방법을 사용하였을 경우는 데이터베이스가 커지면 복잡도 역시 선형적



(a) 상세 검색과 BTC 검색 (b) 차원 축약 검색과 통합 검색  
 그림 21. 데이터베이스 크기 변화에 따른 성능 비교  
 Fig. 21. Performance comparison with respect to database size.

으로 증가하는 반면, BTC와 통합 검색 방법에서는 복잡도가 비선형적으로 증가하며 증가율이 감소한다는 사실을 알 수가 있다. 그림 22의 (a)는 상세 검색과 BTC 검색의 데이터베이스 크기 변화에 따른 성능 변화를 보여준다. 데이터베이스의 크기가 선형적으로 증가해도 BTC 검색 방법은 선형적으로 증가하지 않고 로그 함수의 형태로 증가한다는 사실을 알 수 있다. 그림 22의 (b)는 차원 축약 검색과 통합 검색의 데이터베이스 크기 변화에 따른 성능 변화를 보여준다. 통합 검색 방법에는 BTC 검색이 포함되기 때문에 BTC 검색의 특성을 그대로 나타낸다. 반면, 차원 축약 검색은 데이터베이스 증가에 따른 복잡도 증가율을 크게 감소시킨다. 즉, 직선의 기울기를 감소시키는 역할을 한다. 실험 결과들을 종합해 보면, 차원 축약 검색은 고차원의 특징

벡터 사용에 있어서의 문제점을 해결해 주고, BTC 검색은 대용량 데이터베이스에서의 문제를 해결할 수 있다는 사실을 알 수가 있다.

## VI. 결 론

본 논문에서는 데이터베이스에 저장된 모든 영상, 원 특징정보와 비교하는 상세 검색에서의 적합성을 동일하게 유지하면서도 검색 속도를 월등히 향상시킬 수 있는 통합 검색 시스템을 제안하였다. 통합 검색 시스템은 알고리즘 적으로 서로 종속적이지 않은 차원 축약 2단계 검색 방법과 BTC를 이용한 2단계 검색 방법이 병합되어 이루어져 있다. 차원 축약 2단계 검색 방법은 특징 벡터의 차원 축약을 통하여 정답을 반드시 포함한 후보 영상들만을 대상으로 원 차원 특징 벡터로 재 검색을 하게 된다. 여기서 1단계 검색과 2단계 검색에서 적용되는 사용자 유사도 기준은 동일하지만, 결국은 1단계의 유사도 기준을 어느 정도 낮추어서 검색한다는 의미가 된다. 이진 트리 클러스터링을 이용한 2단계 검색 방법에서는 동적인 클러스터링 과정을 통해 얻어진 각 클러스터의 중심과 1단계 검색을 하여 후보 클러스터를 선정하고 2단계 검색에서는 후보 클러스터 내의 영상을 대상으로 재 검색을 실시한다. 하지만 유사도 보정의 과정이 없이는 적합성을 보장할 수가 없다. 두 검색 방법을 혼합한 통합 검색 시스템은 내용 기반 영상 검색에서 문제가 되는 고차원 특징 벡터의 사용과 대용량 데이터베이스의 적용 부분을 모두 해결할 수 있는 가능성을 제시하였다. 차원 축약 기법을 사용하여 고차원 특징 벡터를 사용하면서도 적합성을 유지하면서 검색 성능을 향상시킬 수 있었고, 이진 트리 클러스터링 기법을 사용하여 데이터베이스의 크기가 증가함에 따라 비례하는 복잡도 문제를 어느 정도 해결할 수 있었다. 또한 통합 검색 방법의 검색 결과를 보면 차원 축약 검색의 부분이 일정 부분 유사도 보정의 역할을 하고 있다는 것을 알 수가 있다. 실험 결과에 의하면 통합 검색은 적합성을 유지하지만, 이러한 상관관계에 대한 증명이나 해석이 쉽지 않기 때문에 본 논문에서는 두 경우에 대한 성능 향상 정도를 모두 제시하였다. 추후 이러한 부분에 대한 연구가 더 진행된다면, 후보 영상들을 선택하는 부분에서 발생하는 불필요한 유사도 계산을 더 줄일 수가 있을 것이다. 그리고 원하는 결과 영상을 정확히 얻어 내면서도 필요한 유사도 계산을 최소화 할 수 있는 색인 구조의 최적화에 대한 방법 제시가 가능할 것이다.

## 참 고 문 헌

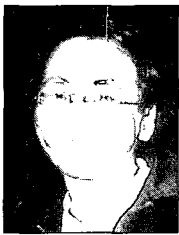
- [1] B.S. Manjunath, Philippe Salembier, Thomas Silora, Introduction to MPEG-7 *Multimedia Content Description Interface*, John Willy & Sons Ltd., 2002.
- [2] <http://www.lgci.com/MIGR/cmip/hmmd/hmmd.html>
- [3] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison Wesley, 1989.
- [4] J. Barros, J. French, W. Martin, P. Kelley, and M. Cannon, "Using the Triangle Inequality to reduce the number of comparisons required for Similarity-based Retrieval," In IS&T/SPIE-Storage and Retrieval for Still Image and Video Databases, volume IV, Jan. 1996.
- [5] Andrew Berman and Linda G. Shapiro, "Selecting Good Keys for Triangle-Inequality-Based Pruning Algorithms," In IEEE International Workshop on Content-based Access of Image and Video Databases, 1997.
- [6] Benjamin Bustos, Gonzalo Navarro, Edgar Chávez, "Pivot Selection Techniques for Proximity Searching in Metric Spaces," In Proc. of the XXI Conference of the Chilean Computer Science Society (SCCC'01), pp 33-40. IEEE CS Press, 2001.
- [7] Vittorio Castelli, Lawrence D. Bergman, *Image Databases Search and Retrieval of Digital Imagery*, John Wiley & Sons Inc., 2002.
- [8] B.V. Bonnländer and A.S. Weigend, "Selecting input variables using mutual information and nonparametric density estimation," Proceedings of International Symposium on Artificial Neural Networks, ISANN94, Tainan, Taiwan, December 15-17, pp. 312- 321, 1994.
- [9] M. Beatty and B.S. Manjunath, "Dimensionality reduction using multidimensional scaling for content-based retrieval," Proceedings of IEEE International Conference Image Processing, ICIP '97 Santa Barbara, Calif., pp. 835-838, October 1997.
- [10] C. Faloutsos and K.-I. Lin, "FastMap: a fast algorithm for indexing, data-mining, and visualization of traditional and multimedia data sets," Proceedings of 1995 ACM SIGMOD International Conference on Management of Data, San Jose, Calif., pp. 163-174, May 1995.
- [11] A. Califano and R. Mohan, "Multidimensional indexing for recognizing visual shapes," Proceedings of IEEE Computer Vision and Pattern Recognition, CVPR '91, pp. 28-34, June 1991.

- [12] H. Wolfson, "Model-based object recognition by geometric hashing," Proceedings of 1st European Conference Computer Vision, Antibes, France, pp. 526-536, April 1990.
- [13] 이근섭, "웨이블릿 변환을 이용한 객체 기반의 이미지 검색 시스템," 한양대학교 석사학위논문, 2001.
- [14] 조정원, 최병욱, "특징벡터의 차원축약 기법을 이용한 2단계 내용기반 이미지검색 시스템," 한국통신학회논문지, 제28권, 제7C호, pp. 719-725. 2003. 7.
- [15] Jung-Won Cho, Seung-Do Jeong, Geun-Seop Lee, Sung-Ho Cho, and Byung-Uk Choi, "A New Fast Image Retrieval Using the Condensed Two-stage Search Method," IEICE Trans. on Commun., Vol. E86-B, No. 12, December 2003.
- [16] 이세한, 이주호, 조정원, "영상 검색의 속도 향상을 위한 차원 축소를 최적화," 대한전자공학회 하계종합학술대회 논문집, 제26권 제 1호, pp. 1515-1518, 2003.
- [17] 조정원, 최병욱, "특징벡터의 클러스터링 기법을 통한 2단계 내용기반 이미지검색 시스템," 전자공학회논문지, 제40권, CI편 제3호, pp. 45-54, 2003. 5.
- [18] Kishan Mehrotra, Chilukuri K. Mohan and Sanjay Ranka, Elements of Artificial Neural Networks, The MIT press, 1997.
- [19] D.E. Rumelhart and D. Zipser., "Feature discovery by competitive learning," Cognitive Science, 9:75-112, 1985.
- [20] L. Xu, A. Krzyzak, and W. Oja., "Rival penalized competitive learning for clustering analysis, RBF net, and curve detection," IEEE trans. on Neural Networks, 4(4):636-649, 1993.
- [21] P.N. Yianilos., "Data structures and algorithms for nearest neighbor search in general metric spaces," Proc. of the 4th Annual ACM-SIAM Symp. on Discrete Algorithms, pp. 311-321, 1993.
- [22] Yong Rui, Thomas S. Huang, and Shih-Fu Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues," Journal of Visual Communication and Image Representation, Vol. 10, no. 4, pp. 39-62, April 1999.
- [23] Alberto Del Bimbo, *Visual Information Retrieval*, Morgan Kaufmann Inc., 1999.

---

 저 자 소 개
 

---



이 세 한 (정회원)  
 2002년 한양대학교 전자통신전과  
 공학과 공학사  
 2004년 한양대학교 전자통신전과  
 공학과 공학석사  
 2004년~현재 LG전자 MC사업본  
 부 단말연구소 연구원  
 <주관심분야 : 내용기반이미지검색, 영상처리>



최 병 욱 (정회원)  
 1973년 한양대학교 전자공학과  
 공학사  
 1978년 일본 경응의숙(KEIO)대학  
 전기공학과 공학석사  
 1981년 일본 경응의숙(KEIO)대학  
 전기공학과 공학박사  
 1981년~현재 한양대학교 정보통신대학 정보통신  
 학부 교수  
 <주관심분야 : 영상처리, 멀티미디어 공학>



조 정 원 (정회원)-교신저자  
 1996년 인천대학교 정보통신공학  
 과 공학사  
 1998년 한양대학교 전자통신공학  
 과 공학석사  
 2004년 한양대학교 전자통신전과  
 공학과 공학박사  
 2004년~현재 제주대학교 컴퓨터교육과 전임강사  
 <주관심분야 : 멀티미디어정보검색, 컴퓨터교육>