

論文2001-38SP-2-1

# 효율적인 영상 검색을 위한 클러스터링 기반 고속 다 해상도 전역 탐색 기법

## (Fast Multi-Resolution Exhaustive Search Algorithm Based on Clustering for Efficient Image Retrieval)

宋秉哲, 金明俊, 羅鍾範

(Byung Cheol Song, Myung Jun Kim, and Jong Beom Ra)

### 요약

유사도 측정자 (similarity measure)에 따라 문의자 (query)의 최적 정합자 (the best match)를 찾는 최적 검색 (optimal retrieval)을 위해서는 데이터베이스의 모든 영상들에 대해 전역 탐색 (exhaustive search)을 수행해야 한다. 그러나, 일반적인 전역 탐색은 방대한 계산량을 요구한다. 그 계산량을 줄이기 위해, 본 논문은 영상 데이터베이스의 클러스터링 (clustering)에 기반한 고속 다 해상도 전역 탐색 기법을 제안한다. 먼저 데이터베이스 내의 모든 영상들을 일정 수의 클러스터 (cluster)들로 나눈다. 각 클러스터는 유사한 특징 (feature)을 갖는 영상들로 구성된다. 그리고, 각 클러스터와 문의자 간 거리 (distance)의 하계 (lower bound)를 구하고, 가능성이 전혀 없다고 판단될 경우 그 클러스터를 제거한다. 가능성이 있다고 판단된 클러스터들에 속한 후보 영상들 중에서 최적 정합자를 찾는다. 또한, 불필요한 특징 정합 연산을 줄이기 위해 다 해상도 데이터 구조에 기반한 거리 부등식 성질 (distance inequality property)을 유도하여, 탐색 과정에 적용한다. 제안한 기법은 고속 다 해상도 전역 탐색 기법으로서 단일 최적 정합자뿐만 아니라 다수의 상위 최적 정합자들도 정확하게 찾을 수 있다. 가장 보편적인 밝기 히스토그램 (luminance histogram)특징을 사용하여, 제안한 기법이 고속의 탐색 속도와 함께 최적 검색을 보장함을 증명해 보인다.

### Abstract

In order to achieve optimal retrieval, i.e., to find the best match to a query according to a certain similarity measure, the exhaustive search should be performed literally for all the images in a database. However, the straightforward exhaustive search algorithm is computationally expensive in large image databases. To reduce its heavy computational cost, this paper presents a fast exhaustive multi-resolution search algorithm based on image database clustering. Firstly, the proposed algorithm partitions the whole image data set into a pre-defined number of clusters having similar feature contents. Next, for a given query, it checks the lower bound of distances in each cluster, eliminating disqualified clusters. Then, it only examines the candidates in the remaining clusters. To alleviate unnecessary feature matching operations in the search procedure, the distance inequality property is employed based on a multi-resolution data structure. The proposed algorithm realizes a fast exhaustive multi-resolution search for either the best match or multiple best matches to the query. Using luminance histograms as a feature, we prove that the proposed algorithm guarantees optimal retrieval with high searching speed.

\* 正會員, 韓國科學技術院 電子電算學科

Science, KAIST)

(Dept. of Electrical Engineering and Computer

接受日字:2000年7月6日, 수정완료일:2000年10月17日

## I. 서론

최근 음성, 정지 영상, 동영상 등 멀티미디어 정보량이 급속히 증가함에 따라, 이들을 색인 (indexing), 저장 (storage), 검색 (retrieval)하기 위한 효율적인 기법들의 필요성이 대두되고 있다<sup>[1-5]</sup>. 특히 디지털 박물관, 인터넷 쇼핑, 의료 영상 검색 시스템 등 다양한 응용 분야가 있는 영상 데이터베이스에서의 효율적인 색인 및 검색에 관한 연구가 활발하다. 효율적인 영상 검색은 범용 멀티미디어 관리 시스템을 위해 반드시 필요하다. 영상을 검색하기 위해서는 먼저 그 영상 내용에 따라 색인이 되어야 한다. 색인은 핵심 단어 (keyword) 들을 이용해 수동적으로 할 수도 있고, 자동으로 추출된 시각 정보를 통해 행해질 수도 있다. 가장 쉬운 방법인 핵심 단어 기반 검색은 소용량 데이터베이스에는 적합하지만<sup>[6]</sup>, 대용량 데이터베이스에서는 그 성능이 크게 떨어진다. 일일이 수작업을 해야 하고, 제한된 수의 핵심 단어들은 각 영상의 자세한 부분까지 표현하기가 힘들기 때문이다.

따라서, 색깔<sup>[7-10]</sup>, 텍스처<sup>[11-12]</sup>, 형태<sup>[13-14]</sup> 같은 특징들을 이용한 내용 기반 영상 검색 (content-based image retrieval)에 관한 연구가 주류를 이루고 있다. 내용 기반 영상 검색 기법들은 문의자와 각 영상 사이의 유사도를 측정하고, 그 순위를 정하여 사용자에게 제공한다. 유사도 측정자가 검색 성능에 미치는 영향은 지대하다. 유사도 측정 방법은 화소 정합 (pixel matching), 특징 정합 (feature matching), 의미론적 유사도 정합 (semantic similarity matching) 등 크게 세 가지로 분류된다. 화소 정합에 의한 유사도는  $L_1$ - 혹은  $L_2$ -놈 (norm) 거리를 구함으로써 얻어진다. 다른 방법에 비해 간단하지만, 영상의 확대, 축소, 회전, 병진 이동 등에 민감하기 때문에 실제 시스템에는 잘 쓰이지 않는다. 의미론적 유사도 정합은 이상적이지만, 아직까지 공감할만한 영상 이해 (image understanding) 기술들이 개발되어 있지 않기 때문에 실제 시스템에 사용하기는 힘들다. 따라서, 본 논문은 색깔이나 윤곽선 같은 특징에 기반한 유사도를 이용한 검색을 다룬다.

유사도 측정자에 따른 문의자의 최적 정합자를 찾기 위해서는 데이터베이스 안의 모든 영상들에 대해 전역 탐색을 수행하여야 한다. 일반적인 전역 탐색 기법 (exhaustive search algorithm; ESA)은 계산량이 많기 때문

에 최근 여러 고속 탐색 기법들이 개발된 바 있다<sup>[15-16]</sup>. Berman과 Shapiro는 탐색 과정에서 최적일 가능성이 전혀 없는 후보들을 적은 계산량으로 제거하기 위해 삼각형 부등식을 도입하였다<sup>[15]</sup>. 추가적인 계산량 감소를 위해 여러 가지 거리 측정자들과 대표 영상 (key image)들을 동시에 사용하기도 했지만, 탐색 속도가 대표 영상들에 따라 변화가 크고, 대용량 데이터베이스에서는 만족할만한 성능을 보이지 못했다. 최근 Triangle Trie라는 데이터 구조를 적용하여 성능을 개선하였지만, 탐색 속도는 여전히 대표 영상들, Trie 깊이, 문턱값 등에 많은 영향을 받는다<sup>[16]</sup>. Shepherd 등은 보조적인 색인 구조로서 공간 기입 곡선 (space-filling curve) 집합에 기반한 기법을 제안하였다. 이 기법은 다 차원 특징 벡터들을 통해 접속 가능한 영상 데이터베이스에 적합하다<sup>[17]</sup>. 그러나, 최적 검색을 보장할 수 없을 뿐만 아니라, 적절한 검색 정확도 (retrieval accuracy)를 얻기 위해서는 많은 수행 시간이 요구되는 단점이 있다. 한편, Krishnamachari과 Mottaleb는 데이터베이스에 있는 영상들을 계층적 클러스터링 기법으로 비슷한 색깔을 갖는 클러스터들로 분할하는 클러스터 기반 색인 기법을 새롭게 제안하였다<sup>[18]</sup>. 탐색 과정에서 문의자 영상을 데이터베이스의 모든 영상들과 비교하지 않고, 클러스터링에 의해 일부 영상들과 비교하기 때문에 계산량을 많이 줄일 수 있다. 클러스터 기반 기법은 원하는 정확도를 얻기 위한 비교 횟수가 데이터베이스 크기와 선형적으로 비례하지 않기 때문에 대용량 데이터베이스에 적합하다. 그러나, 기존 클러스터 기반 기법들은 최적 검색을 보장하지 못하며, 그 탐색 속도가 만족할만한 검색 정확도를 얻기에 충분히 빠르지 않다.

본 논문은 고속 최적 검색을 위한 고성능 클러스터 기반 다 해상도 탐색 기법을 제안한다. 먼저 데이터베이스 내의 모든 영상들이 K-평균 클러스터링 기법에 의해 비슷한 특징을 갖는 일정 수의 클러스터들로 분할된다. 제안한 기법은 두 가지 부등식 성질 (property)들을 이용한다. 첫 번째 성질은 각 클러스터 내의 영상들과 문의자 간 거리의 하계 (lower bound)를 제시한다. 이 성질은 탐색 과정에서의 불필요한 비교 연산을 줄이는데 기여한다. 두 번째 성질은 다 해상도 특징 공간에서 인접한 계층의 거리들 간 상관 관계를 보인다. 이 성질 역시 탐색 과정에서 불필요한 비교 연산을 줄이기 위해 이용된다. 제안한 기법은 위 두 가지 성질을 이용하여 기존의 어떤 탐색 기법들보다 빠르게 최적

정합자를 찾는다. 또한, 다수의 상위 최적 정합자들도 정확하고 빠르게 찾는다. 모의 실험 결과는 제안한 기법이 대용량 영상 데이터베이스에서의 고속 최적 검색에 적합함을 증명해준다.

본 논문의 구성은 다음과 같다. II절에서는, 탐색 속도 향상을 위한 두 부등식 성질들을 유도하고, 이 두 가지 성질들을 이용한 새로운 클러스터 기반 고속 다해상도 전역 탐색 기법을 제안한다. 첫 번째 성질은 가능성이 전혀 없는 클러스터들을 정확하게 제거하기 위해 이용되며, 두 번째 성질은 다해상도 데이터 구조에 기반하여 비교 연산을 줄이기 위해 이용된다. III절에서는 다양한 실험 결과를 통해 제안한 방법의 우수성을 입증하고, 마지막 IV절에서 결론을 맺는다.

## II. 클러스터 기반 다해상도 탐색 기법

클러스터 기반 탐색 기법들은 대용량 영상 데이터베이스에 적합하다고 알려져 있다<sup>[18]</sup>. 그러나, 서론에서 언급된 것처럼, 기존의 클러스터 기반 기법들은 최적 검색을 보장할 수 없다. 따라서 이 절에서는 클러스터화된 데이터베이스에서 최적 검색을 보장할 수 있는 성질 (property)을 유도하고, 이 성질에 이용한 고속 다해상도 전역 탐색 기법을 제안한다. 본 논문에서는 편의상 밝기 히스토그램 특징만을 고려한다.

### 1. 데이터베이스의 클러스터링

사용할 데이터베이스를  $I = \{I_1, \dots, I_i, \dots, I_N\}$ 라 하자.  $N$ 은 데이터베이스 내 영상들의 개수를 의미한다. 먼저, 비슷한 밝기를 갖는 영상들이 하나의 클러스터를 이루도록 MacQueen  $K$ -평균 클러스터링<sup>[19]</sup>을 이용하여, 정해진 수의 클러스터들로 데이터베이스를 분할한다.  $K$ 개의 클러스터들은 각자의 평균 중심점을 갖는다. 클러스터링 연산은 실제 탐색 전에 이루어지므로, 이를 위한 시간은 탐색 시간에 포함되지 않는다. 데이터베이스의 클러스터링은 다음과 같이 이루어진다.

- 1) 클러스터의 수 ( $K; K < N$ )를 정한다.
- 2) 클러스터 중심점들의 밝기 히스토그램들,  $C_1, C_2, \dots, C_K$ 을 초기화한다. 데이터베이스에서  $K$ 개 영상들을 임의로 선택한다. 효율적인 초기화를 위해, 두 중심점들간의 최소 거리가  $\theta$ 를 넘지 않도록 제한한다.
- 3) 초기 중심점들을 제외한 나머지 영상들 각각에 대해 가장 가까운 클러스터 중심점을 찾아 해당 클러스터에

포함시키고, 다음 식에 의해 중심점을 갱신한다.

$$C_k = \frac{1}{n(\Phi_k) + 1} [n(\Phi_k)C_k + X_i] \quad (1)$$

여기서,  $X_i$ 는 클러스터  $k$ 에 추가될  $i$ 번째 원소이며,  $\Phi_k$ 는 클러스터  $k$ 를,  $n(\Phi_k)$ 는  $\Phi_k$ 에 속한 원소들의 개수이다.

4) 모든 원소들에 대해 수렴할 때까지 이 단계 3을 반복한다.

최종적으로, 중심점 집합  $I^0 = \{C_1, C_2, \dots, C_K\}$ 이 얻어진다.

### 2. 클러스터화된 데이터베이스에서의 최적 탐색을 위한 클러스터 제거

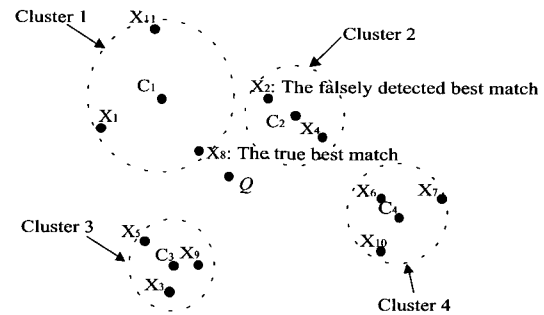


그림 1. 기존 클러스터 기반 기법들의 본질적인 문제점. 여기서,  $N=11, K=4$

Fig. 1. Inherent problem of conventional cluster-based algorithms. In this example,  $N=11$  and  $K=4$ .

그림 1은 기존의 클러스터 기반 기법들의 본질적인 문제점을 지적하고 있다. 중심점들 중에서  $C_2$ 가 문의자  $Q$ 에 가장 가깝기 때문에 클러스터 2를 후보로 선택한다. 클러스터 2에 속한 각 원소와  $Q$ 와의 거리를 계산하여,  $X_2$ 를 최적 정합자로 선택한다. 그러나, 실제 최적 정합자는 클러스터 1의  $X_8$ 이다. 실제 최적 정합자가 속한 클러스터의 중심점이 항상  $Q$ 와 가장 가깝지는 않기 때문에 이런 문제가 발생한다. 이 문제를 해결하기 위해,  $Q$ 와 가까운 여러 개의 클러스터들을 동시에 탐색하는 방법이 시도되었지만<sup>[18]</sup>, 여전히 최적 검색은 보장하지 못했다.

본 논문은 위 문제의 해법을 제시한다. 먼저,  $Q$ 와 가장 가까운 클러스터에서 초기 최소 거리  $d_{\min}$ 를 구한다.

$$d_{\min} = \min_{X_i \in \Phi_{k_{\min}}} d(X_i, Q) \quad (2)$$

식 (2)에서

$$C_{k_{\min}} = \arg \min_{C_k \in \Pi^0} d(C_k, Q) \quad (3)$$

$d(X, Y)$ 는 두 히스토그램  $X$ 와  $Y$ 의  $L_1$ -놈 거리이며,  $\Phi_{k_{\min}}$ 는 식 (3)에서  $C_{k_{\min}}$ 에 대응하는 클러스터 집합을 의미한다. 그림 1에서  $d_{\min}$ 는  $d(X_2, Q)$ 이다. 그리고  $\Phi_{k_{\min}}$ 를 제외한 각 클러스터 내에서 중심점과 가장 먼 원소를 찾고, 그 거리를 다음과 같이 정의한다.

$$\delta_k = \max_{X_i \in \Phi_k} d(X_i, C_k) \quad (4)$$

클러스터  $k$ 의  $\delta_k$ 는 미리 계산되어 저장된다.  $d_{\min}$ 와  $\delta_k$ 값들을 이용하여, 각 클러스터가 최적 검색을 위해 탐색될 필요가 있는지 없는지를 결정한다. 이 결정을 위한 성질은 다음과 같다:

**성질 1:**

$$\text{If } d(C_k, Q) - \delta_k > d_{\min}, \min_{X_i \in \Phi_k} d(X_i, Q) > d_{\min} \quad (5)$$

**증명:**

$$X_{i_{\min}} = \arg \min d(X_i, Q) \quad (6)$$

삼각형 부등식에 의해,

$$d(X_{i_{\min}}, Q) \geq d(C_k, Q) - d(X_{i_{\min}}, C_k) \quad (7)$$

식 (4)로부터,

$$\delta_k = \max_{X_i \in \Phi_k} d(X_i, C_k) \geq d(X_{i_{\min}}, C_k) \quad (8)$$

다음 부등식이 식 (7)과 (8)로부터 얻어진다.

$$d(X_{i_{\min}}, Q) \geq d(C_k, Q) - d(X_{i_{\min}}, C_k) \geq d(C_k, Q) - \delta_k \quad (9)$$

만약  $d(C_k, Q) - \delta_k > d_{\min}$ 이면,

$$d(X_{i_{\min}}, Q) = \min_{X_i \in \Phi_k} d(X_i, Q) > d_{\min} \quad (10)$$

따라서, 식 (5)가 성립한다.

증명 끝.

성질 1에서,  $d(C_k, Q) - \delta_k$ 은  $Q$ 와 클러스터  $k$ 내의 원소 사이 거리의 하계를 의미한다 (그림 2). 만약  $d(C_k, Q) - \delta_k$ 이  $d_{\min}$ 보다 크면, 클러스터  $k$ 에는  $d_{\min}$ 보다 작은 거리를 갖는 원소가 존재하지 않으므로 더 이상 클러스터  $k$ 를 고려할 필요가 없다. 따라서, 성질 1에 조건을 적용하여, 효과적으로 가능성이 전혀 없는 클러스터들을 정확하게 모두 제거할 수 있다. 그러나, 가장 가까운 클러스터 찾는 과정과 최적 정합자를 찾는 과정은 여전히 많은 계산량을 요구한다. 이 계산량을 줄이기 위해, 다 해상도 데이터 구조에 기반한 또다른 부등식 성질을 유도하고, 이를 이용한 고속 최적 탐색을 제안한다.

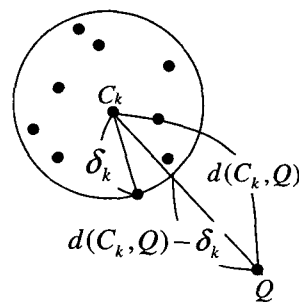


그림 2. 각 클러스터에 대한 거리 부등식 특징. 클러스터  $k$ 의 각 원소와 문의자 간 거리는  $d(C_k, Q) - \delta_k$ 보다 작을 수 없다. 여기서  $\delta_k$ 는 중심점과 각 영상들 간 거리의 최대값이다.

Fig. 2. Distance inequality feature in a cluster. The distance of any image in the  $k$ -th cluster from the query cannot be less than  $d(C_k, Q) - \delta_k$ . Here  $\delta_k$  is the maximum distance of images from the cluster center.

3. 다 해상도 데이터베이스에서의 고속 전역 탐색

1) 다 해상도 데이터 구조:  $B(B=2^L)$ 빈을 갖는 정규화된 밝기 히스토그램 특징에 대한 다 해상도 데이터 구조를 고려한다. 히스토그램  $X$ 의 다 해상도 구조는 히스토그램열  $\{X^0, X^1, \dots, X^l, \dots, X^L\}$ 로 정의할 수 있다. 여기서,  $X=X^L$ 이다.  $X^l$ 는  $2^l$ 개의 빈을 가지며,  $X^{l+1}$ 로부터 1/2비율로 해상도를 줄임으로써 얻어진다 (그림 3). 현재 계층의 각 화소값은 인접한 상위 계층

의 두 화소값을 더하여 얻어진다. 즉,  $X^l(m)$ 이  $X^l$ 의  $m$ 번째 빈 값이라 할 때,  $X^l(m)$ 은 다음과 같이 얻어진다:

$$X^l(m) = X^{l+1}(2m-1) + X^{l+1}(2m), \quad 1 \leq m \leq 2^l. \quad (11)$$

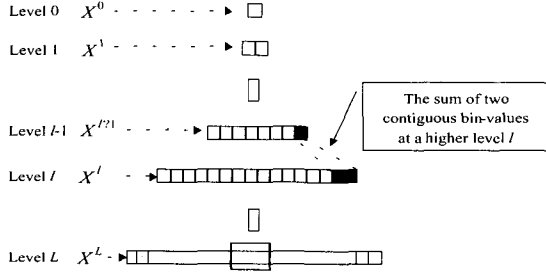


그림 3.  $2^L$ 빈을 갖는 밝기 히스토그램  $X$ 의 다 해상도 구조.

Fig. 3. Multi-resolution structure of a luminance histogram  $X$  having  $2^L$  bins.

2) 다 해상도 데이터 구조에서의 부등식 성질: 유도할 성질의 기본 개념은 동영상 압축을 위한 고속 움직임 추정 기법 중 하나인 SEA기법에서 사용된 바 있다<sup>[20]</sup>. 주어진 탐색 영역 내에서 가능성이 전혀 없는 후보 움직임 벡터들을 제거하기 위해, SEA는 현재 블록과 이전 프레임의 각 후보 블록 간 차이의 하계가 블록합 (block sum)간  $L_1$ -놈 차이 수 있다는 점을 이용하여, 계산량을 감소시켰다. 이 개념을 다 해상도 특징 공간으로 확장하여, 다음과 같은 성질을 유도할 수 있다

**성질 2:**

$$d(X, Y) \equiv d^L(X, Y) \geq d^{L-1}(X, Y) \geq \dots \geq d^1(X, Y) \geq d^0(X, Y) \quad (12)$$

여기서,  $d^l(X, Y)$ 는 계층  $l$ 에서의 두 히스토그램  $X$ 와  $Y$ 간  $L_1$ -놈 거리, 즉  $d(X^l, Y^l)$ 을 의미한다.

**증명:**

계층  $l+1$ 에서 두 히스토그램  $X$ 와  $Y$ 의  $L_1$ -놈 차는 다음과 같이 구해진다.

$$d^{l+1}(X, Y) = \sum_{m=1}^{2^{l+1}} |X^{l+1}(m) - Y^{l+1}(m)| \\ = \sum_{m=1}^{2^l} (|X^{l+1}(2m-1) - Y^{l+1}(2m-1)| + |X^{l+1}(2m) - Y^{l+1}(2m)|) \quad (13)$$

$X^{l+1}$ 와  $Y^{l+1}$ 는 각각  $2^{l+1}$ 개의 빈을 가지며,  $X^{l+1}(m)$ 은  $X^{l+1}$ 의  $m$ 번째 빈의 값이다.

한편, 식 (11)을 사용하여, 계층  $l$ 에서의 거리를 다음과 같이 표현할 수 있다.

$$d^l(X, Y) = \sum_{m=1}^{2^l} |X^l(m) - Y^l(m)| \\ = \sum_{m=1}^{2^l} |X^{l+1}(2m-1) - Y^{l+1}(2m-1) + X^{l+1}(2m) - Y^{l+1}(2m)| \quad (14)$$

$|A+B| \geq |A| + |B|$ 이므로,

$$\sum_{m=1}^{2^l} (|X^{l+1}(2m-1) - Y^{l+1}(2m-1)| + |X^{l+1}(2m) - Y^{l+1}(2m)|) \\ \geq \sum_{m=1}^{2^l} (|X^{l+1}(2m-1) - Y^{l+1}(2m-1)| + |X^{l+1}(2m) - Y^{l+1}(2m)|) \quad (15)$$

식 (13), (14), (15)로부터,

$$d^l(X, Y) = \sum_{m=1}^{2^l} (|X^{l+1}(2m-1) - Y^{l+1}(2m-1)| + |X^{l+1}(2m) - Y^{l+1}(2m)|) \\ \geq \sum_{m=1}^{2^l} (|X^{l+1}(2m-1) - Y^{l+1}(2m-1)| + |X^{l+1}(2m) - Y^{l+1}(2m)|) \\ = d^l(X, Y) \quad (16)$$

식 (16)으로부터 식 (12)가 성립한다.

증명 끝.

정규화된 밝기 히스토그램의 경우  $X^0$ 와  $Y^0$  모두 1이기 때문에,  $d^0(X, Y)$ 은 항상 0이다. 성질 2에 따르면  $d^l(X, Y)$ 이 특정 값보다 클 경우,  $d^L(X, Y)$ 은 항상 그 특정 값보다 크다. 상위 계층에서의 거리 계산이 하위 계층에서의 거리 계산보다 많은 연산을 필요로 하기 때문에, 위 성질을 탐색 과정에 적용하여 하위 계층들에서 가능성이 전혀 없는 후보들을 많이 제거함으로써 계산량을 크게 줄일 수 있다.

3) 다 해상도 데이터베이스에서의 고속 전역 탐색:  $\Omega^0 = \{X_1, \dots, X_i, \dots, X_N\}$ 는 데이터베이스 내 영상들의 히스토그램 집합이다. 각 영상의 다 해상도 히스토그램은 미리 계산되어 저장된다. 성질 2에 기반한 고속 다 해상도 전역 탐색 기법 (MSAs)을 요약하면 다음과 같다:

- 1)  $Q$ 의 다 해상도 히스토그램을 구한다.
- 2) 초기  $d_{\min}$ 을 무한대로 설정한다.
- 3)  $i$ 와  $l$ 는 모두 1로 한다.
- 4)  $l=L$ 이면, 단계 6으로 간다. 만약  $i$ 가  $N$ 보다 크면, 단계 7로 간다.
- 5)  $d^l(X_i, Q)$ 을 구한다.  $d^l(X_i, Q) > d_{\min}$ 이면, 현재 후

보  $X_i$ 를 제거하고,  $i$ 와  $l$ 을 각각  $i+1$ 과  $l$ 로 갱신한다. 그렇지 않으면,  $l$ 을  $l+1$ 로 갱신한 후 단계 3으로 간다.

6)  $d^l(X_i, Q) > d_{\min}$ 이면, 현재 후보  $X_i$ 를 제거한다. 그렇지 않으면,  $d_{\min}$ 을  $d^l(X_i, Q)$ 로 갱신한다.  $i$ 와  $l$ 을 각각  $i+1$ 과  $l$ 로 갱신한 후 단계 4로 간다.

7) 최종  $d_{\min}$ 을 갖는 영상을 최적 정합자로 선택한다.

4) 다 해상도 데이터 구조 구성을 위한 계산량: 데이터베이스의 각 영상의 다 해상도 히스토그램은 미리 계산되어 저장된다. 그러나, 문의자 영상의 다 해상도 히스토그램은 탐색 시간동안 얻어져야 하므로, 그 계산량을 고려해야만 한다. 정규화된 다 해상도 히스토그램을 얻기 위해서는 단지 254번의 덧셈 연산만이 요구된다. 1회의 정합 과정에서 511번의 덧셈 연산과 256번의 절대값 연산이 소요됨을 감안할 때, 다 해상도 밝기 히스토그램 형성에 드는 계산량은 무시할 만하다. 한편, 다 해상도 히스토그램들을 저장하기 위해 추가적인 메모리가 필요하나, 히스토그램의 크기가 영상 데이터 크기보다 훨씬 작기 때문에 그 추가 메모리 증가 역시 무시할 만하다.

#### 4. 클러스터 기반 다 해상도 탐색 기법 (CMSA)

II.2절의 클러스터 제거 조건과 II.3절의 다 해상도 탐색 기법을 융합함으로써, 고속 최적 탐색을 위한 새로운 클러스터 기반 다 해상도 탐색 기법 (cluster-based multi-resolution search algorithm; CMSA)을 제안한다. 문의자가 주어지면, MSA<sub>S</sub>를 통해 가장 근접한 클러스터 중심점을 찾고, 그 클러스터 내에서 초기 최적 정합자,  $d_{\min}$ 를 구한다. 그리고, 성질 1의 클러스터 제거 조건을 적용한다. 가능성이 있다고 판단된 클러스터들에 대해 MSA<sub>S</sub>를 적용하여 최적 정합자 (들)를 찾는다.

가장 가까운 클러스터 중심점을 찾을 때 MSA<sub>S</sub>가 사용되기 때문에, 클러스터 제거 과정에서 모든  $d^l(C_k, Q)$  값들이 존재하지는 않는다. 이 보다 크면,  $l$ 보다 큰 계층들에서의 거리들  $d^{l+1}(C_k, Q), \dots, d^L(C_k, Q)$ 은 계산되지 않기 때문이다. 결국, 식 (5)의 성질 1을 클러스터 제거 과정에 도입하기 위해 다시  $d^l(C_k, Q)$ 을 계산해야 하는 문제점이 생긴다. 그래서,  $d(C_k, Q) \equiv d^L(C_k, Q) \geq d^l(C_k, Q)$ 의 관계를 이용해 성질 1을 다음

과 같이 변형시킨다.

성질 1.1:

$$\text{If } d^{l_k}(C_k, Q) - \delta_k > d_{\min}, \min_{X_i \in \Phi_i} d(X_i, Q) > d_{\min} \quad (17)$$

여기서,  $l_k \leq L$ .

성질 1.1에 따르면,  $d^{l_k}(C_k, Q) - \delta_k$ 이  $d_{\min}$ 보다 클 경우, 클러스터  $k$ 를 제거할 수 있다. 반대로  $d^{l_k}(C_k, Q) - \delta_k$ 이  $d_{\min}$ 보다 작으면, 최적 정합자가 그 클러스터에 존재할 가능성이 있으므로, 클러스터  $k$ 를 탐색한다. 모든  $k$ 에 대해  $d^{l_k}(C_k, Q)$ 과  $\delta_k$ 는 이미 알려져 있으므로, 이 결정을 위한 추가적인 계산량은 없다.

본 논문은 위 성질들을 바탕으로 출력 최적 정합자 수에 따라 다음 두 가지 CMSA를 설계한다. 첫째는 하나의 최적 정합자를 출력하는 CMSA이며, 둘째는 다수의 상위 최적 정합자들을 출력하는 CMSA이다.

1) 하나의 최적 정합자를 출력할 경우의 CMSA (CMSA<sub>S</sub>): CMSA<sub>S</sub>는 크게 세 단계로 구성된다. 먼저,  $C_{k_{\min}}$ 을 MSA<sub>S</sub>로 찾는다. 그리고, 초기  $d_{\min}$ 을  $\Phi_{k_{\min}}$ 에서 찾는다. 마지막으로, 성질 1.1에 의해 선택된 후보 클러스터들에만 MSA<sub>S</sub>를 다시 적용하여 최적 정합자를 찾는다. CMSA<sub>S</sub>의 탐색 과정을 요약하면 다음과 같다.

#### CMSA<sub>S</sub> :

1) 문의자  $Q$ 과  $C_k$ 에 대해 MSA<sub>S</sub>를 수행하여, 최소 거리  $d_{\min}'$ 를 갖는 클러스터  $k_{\min}$ 을 찾는다.

2) 초기  $d_{\min}$ 을  $d_{\min}'$ 라고 하고, MSA<sub>S</sub>를  $\Phi_{k_{\min}}$ 에 적용하여  $d_{\min} = \min_{X_i \in \Phi_i} d^L(X_i, Q)$ 을 구한다.

3-1)  $k$ 를 1로 한다.

3-2)  $k = k_{\min}$ 이면,  $k$ 를  $k+1$ 로 갱신한다.  $k$ 가  $K$ 보다 크면, 단계 3-4으로 간다.

3-3)  $d^{l_k}(C_k, Q) - \delta_k$ 이  $d_{\min}$ 보다 크면, 클러스터  $k$ 를 제거한다. 그렇지 않으면,  $\Phi_k$ 에 대해 MSA<sub>S</sub>를 사용하여,  $d_{\min}$ 을 갱신한다.  $k$ 를  $k+1$ 로 갱신한 후, 단계 3-2로 간다.

3-4) 최종  $d_{\min}$ 를 갖는 영상을 최적 정합자로 선택한다.

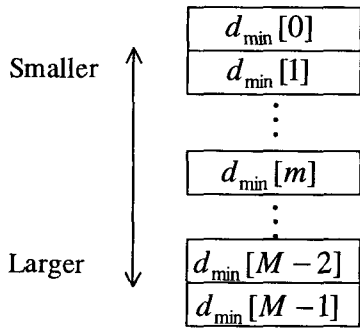


그림 4. 최소 거리 배열. 여기서  $d_{\min}[m-1]$ 에는  $m$  번째 최소 거리가 저장된다.  
 Fig. 4. Minimum distance array. Here,  $d_{\min}[m-1]$  corresponds to the  $m$ -th smallest distance.

2) 다수의 최적 정합자를 출력할 경우의 CMSA (CMSA<sub>M</sub>): 먼저, CMSA<sub>S</sub>와 동일한 방법으로  $C_{k_{\min}}$ 을 찾는다. 그리고, 상위  $M$ 개의 최적 정합자들의 거리 값들을 저장하기 위한 그림 4와 같은 배열을 다음 규칙에 따라 채운다.

- $n(\Phi_{k_{\min}}) \geq M$ 이면,  $\Phi_{k_{\min}}$  내의 상위  $M$  최적 정합자들을 작은 거리 값 순서로 배열에 채운다.
- $n(\Phi_{k_{\min}}) < M$ 이면,  $\Phi_{k_{\min}}$  내 모든 원소의 거리들을 계산되고, 그 값들을 작은 값 순서로 배열에 저장하고, 나머지는 무한대 값을 채운다.

$\Phi_{k_{\min}}$  안의 상위  $M$  최적 정합자들을 찾을 수 있는 MSA<sub>M</sub>를 MSA<sub>S</sub>를 변형하여 제안한다. MSA<sub>M</sub>를 요약하면 다음과 같다.

- 1) 문의자  $Q$ 의 다 해상도 히스토그램을 구한다.
- 2)  $d_{\min}[\cdot]$  안의 모든 원소들을 무한대 값으로 초기화한다.
- 3)  $i$ 과  $l$ 을 모두 1로 한다.
- 4)  $l=L$ 이면, 단계 6으로 간다.  $i$ 가  $n(\Phi_{k_{\min}})$ 보다 크면, 단계 7로 간다.
- 5)  $d^l(X_i, Q)$ 을 계산한다.  $d^l(X_i, Q)$ 이  $d_{\min}[M-1]$ 보다 크면, 현재 후보  $X_i$ 를 제거하고,  $i$ 과  $l$ 를 각각  $i+1$ 과 1로 갱신한 후 단계 3으로 간다. 그렇지 않으면,  $l$ 을  $l+1$ 로 갱신한 후 단계 3으로 간다.
- 6)  $d^l(X_i, Q)$ 이  $d_{\min}[m-1]$ 보다 크면, 현재 후보  $X_i$ 를 제거한다. 그렇지 않으면,  $d_{\min}[m-1]$ 을  $d^l(X_i, Q)$ 로 갱신한다. 그리고,  $d_{\min}[\cdot]$ 을 작은 값 순서로 정렬한다.  $i$ 과  $l$ 을 각각  $i+1$ 과 1로 갱신한 후

단계 4로 간다.

- 7) 최종  $d_{\min}[\cdot]$ 에 있는  $M$ 개의 영상들을 최상위  $M$  최적 정합자들로 선택한다.

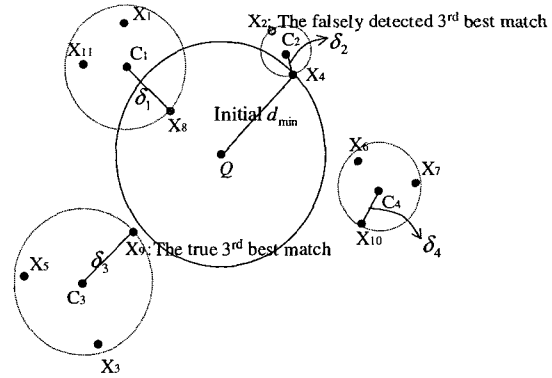


그림 5. 그릇된 판정의 예. 여기서,  $M=3, K=4, N=11$ .  
 Fig. 5. An example of false decision. Here,  $M=3, K=4, N=11$ .

MSA<sub>M</sub>를  $\Phi$ 에 적용하여  $d_{\min}[\cdot]$ 을 채운 후, 나머지 클러스터들 중 성질 1.1에 따라 선택된 각 클러스터에 MSA<sub>M</sub>를 다시 적용하는 방식으로  $d_{\min}[\cdot]$ 을 갱신한다. 최종  $d_{\min}[\cdot]$ 에 대응하는 영상들을 상위  $M$  최적 정합자들로 선택한다. 그러나, 이 탐색 기법으로는 실제 상위  $M$  최적 정합자들을 정확하게 찾지 못할 경우도 있다. 그림 5의 예에서 알 수 있듯이,  $X_8, X_4, X_2$ 가 상위 3개의 최적 정합자들로 선택되었다. 그러나, 실제 3번째 최적 정합자는  $X_9$ 가 아니라  $X_9$ 이다. 따라서, 다음과 같이  $d_{\min}$ 를  $d_{\min}[M-1]$ 로 대체하여 성질 1.1에서 완화된 클러스터 제거 조건을 유도하고 탐색의 후처리 과정에 적용한다.

성질 1.2:

$$\text{If } d^l(C_k, Q) - \delta_k > d_{\min}[M-1], \min_{X_i \in \Phi_i} d(X_i, Q) > d_{\min}[M-1] \quad (18)$$

성질 1.2을 이용하면, 상위  $M$  최적 정합자들을 정확히 찾을 수 있다. 최종적으로 CMSA<sub>M</sub>를 요약하면 다음과 같다.

**CMSA<sub>M</sub>:**

- 1) CMSA<sub>S</sub>의 단계 1과 동일하게, 최소 거리  $d_{\min}'$ 을

갖는 클러스터  $k_{\min}$  을 찾는다.

2)  $n(\Phi_{k_{\min}}) \geq M$  이면,  $MSA_M$ 에 의해 상위  $M$  최적 정합자들이 찾아, 그 거리값들을  $d_{\min}[\cdot]$ 에 저장한다.  $n(\Phi_{k_{\min}}) \geq M$  이면,  $n(\Phi_{k_{\min}})$ 개의 거리값들이 작은 값 순서로  $d_{\min}[\cdot]$ 에 저장하며, 배열의 나머지는 무한대값으로 채운다.

3-1)  $k$ 를 1로 한다.

3-2)  $k = k_{\min}$  이면,  $k$ 를  $k+1$ 로 갱신한다.  $k$ 가  $K$ 보다 크면, 단계 3-5으로 간다.

3-3)  $d^h(C_k, Q) - \delta_k$ 이  $d_{\min}[0]$ 보다 크면, 클러스터  $k$ 를 제거하고,  $k$ 를  $k+1$ 로 갱신한 후 단계 3-2로 간다.

3-4)  $MSA_M$ 을  $\Phi_k$ 에 적용함으로써,  $d_{\min}[\cdot]$ 을 갱신

한다.  $k$ 를  $k+1$ 로 갱신한 후 단계 3-2로 간다.

3-5)  $k$ 를 1로 한다.

3-6) 클러스터  $k$ 가 단계 3-4에서 이미 조사되었으면,  $k$ 를  $k+1$ 로 갱신한다.  $k$ 가  $K$ 보다 크면, 단계 3-9로 간다.

3-7)  $d^h(C_k, Q) - \delta_k$ 이  $d_{\min}[M-1]$ 보다 크면, 클러스터  $k$ 를 제거하고  $k$ 를  $k+1$ 로 갱신한 후 단계 3-6으로 간다.

3-8)  $MSA_M$ 를  $\Phi_k$ 에 적용함으로써,  $d_{\min}[\cdot]$ 를 갱신한다.  $k$ 를  $k+1$ 로 갱신한 후 단계 3-6으로 간다.

3-9) 최종  $d_{\min}[\cdot]$ 에 대응하는  $M$ 개의 영상들을 최적 정합자들로 선택한다.

$d_{\min}[\cdot]$ 를 갱신할 때 성질 1.2에 기반한 단계 3-5부터 3-8까지를 처음부터 탐색에 적용하지 않고, 단계



그림 6. 그영상 데이터베이스에 포함된 영상 샘플들. 다양한 종류의 영상들이 포함되어 있다.

Fig. 6. Some samples from the image database. Note that various kinds of images are used for the database.



3-1부터 3-4까지를 먼저 적용함에 주목할 필요가 있다. 이는 통계적으로 최적 정합자들의 대부분이 단계 3-1부터 3-4에서 보다 빠르게 찾아지기 때문에, 전체 탐색 속도를 크게 향상시킬 수 있기 때문이다.

### III. 모의 실험 결과

#### 1. 실험 조건

데이터베이스는 10,000개의 정지 영상들로 이루어져 있다 ( $N = 10,000$ ). 이 중 7,000개의 영상들은 MPEG-7 데이터 세트에서 추출하였으며<sup>[21]</sup>, 3,000개의 영상들은 ftp를 통해 얻은 영상들이다<sup>[22]</sup>. 데이터베이스가 특정 부류의 영상에 치우치지 않도록, 자연, 건물, 사람 등 다양한 종류의 영상들로 데이터베이스를 구성하였다. 그림 6에 데이터베이스에 있는 일부 영상들을 보였다. 데이터베이스를  $K$ 개의 클러스터들로 분할할 때,  $\theta$ 는 실험적으로 0.5로 정하였다.  $K=500, 1000, 1500$  등 세 가지 경우에 대해 제안한 CMSA를 평가하였다. 데이터베이스에 포함되지 않은 100개의 서로 다른 특성들을 지닌 정지 영상들을 테스트 영상으로 사용하였다. 실험은 167MHz CPU와 64MByte RAM을 갖는 SUN Ultra1 워크스테이션에서 수행되었다. 성능 평가 측정자로서 CPU 실행 시간에 따른 속도 상승률 (speed-up ratio: SUR)을 도입하였으며, SUR은 다음과 같이 정의된다.

$$SUR = \frac{T_{ESA}}{T_{COMP}} \quad (19)$$

여기서  $T_{ESA}$ 와  $T_{COMP}$ 는 각각 ESA의 실행 시간과 비교되는 기법의 실행 시간을 의미한다. 공정한 비교 평가를 위해 100개의 실험 영상들에 대해 다음 세 가지의 종류의 SUR을 측정했다: 평균 SUR ( $SUR_{AVG}$ ), 최대 SUR ( $SUR_{MAX}$ ), 그리고 최소 SUR ( $SUR_{MIN}$ ). 제안한 기법이 최적 검색을 보장하기 때문에 탐색 정확도는 측정하지 않았다.

#### 2. CMSA<sub>S</sub>의 성능 평가

CMSA<sub>S</sub>를 Berman과 Shapiro의 삼각형 기반 기법 (triangle inequality-based algorithm: TIA)<sup>[15]</sup>과 비교했다. TIA의 구현 시 데이터베이스에서 임의로 30개의 키 영상들을 추출했다. 이와 함께, 데이터베이스의 클러스터링 효과를 보기 위해,  $K=1$ 인 경우에 대해서도 실험하였다. 표 I은 그 비교 결과를 보여준다.  $K$ 가

500~1500일 때, CMSA<sub>S</sub>는 평균적으로 ESA보다 40배 가량 빠르며, TIA에 비해서는 약 7.5배 빠름을 알 수 있다. 또한, 클러스터화되지 않은 영상 데이터베이스 ( $K=1$ )와 비교할 때, 클러스터화된 영상 데이터베이스에서 CMSA<sub>S</sub>의 속도가 평균 65%정도 빨라짐을 알 수 있다. 즉, 데이터베이스를 클러스터링함으로써 탐색 속도를 크게 향상시킬 수 있다. CMSA<sub>S</sub>의  $SUR_{MIN}$ 이 항상 TIA의  $SUR_{MAX}$ 보다 크다는 사실도 주목할 만하다.

표 1. 여러  $K$ 에 대한 CMSA<sub>S</sub>과 TIA와의 SUR 비교

Table 1. SUR comparison of CMSAS with the existing TIA for various  $K$ 's.

	CMSA <sub>S</sub>				TIA
	$K=1$	$K=500$	$K=1000$	$K=1500$	
$SUR_{AVG}$	25.9	38.2	40.1	40.9	5.5
$SUR_{MAX}$	53.0	159	159	159	12.3
$SUR_{MIN}$	15.5	18.7	19.9	19.9	1.5

표 2. CMSA<sub>S</sub>의 각 계층에서 조사되는 후보들의 비율

Table 2. Percentages of the candidates that are examined at each level of CMSAS.

	$P^l_{CMSA_S}(\%)$			
	$K=1$	$K=500$	$K=1000$	$K=1500$
$l=1$	100	69.0	61.5	57.1
$l=2$	39.6	28.5	26.0	24.5
$l=3$	12.5	8.6	8.3	8.0
$l=4$	4.0	1.9	1.9	1.9
$l=5$	1.6	0.51	0.52	0.53
$l=6$	0.89	0.22	0.23	0.23
$l=7$	0.66	0.14	0.15	0.16
$l=8$	0.39	0.1	0.1	0.1

표 II는 CMSA<sub>S</sub>가 각 계층에서 얼마나 많은 후보 영상들을 조사하는지 보여준다. 각 계층에서 조사되는 후보들의 비율이 하나의 성능 비교 측정자로서 도입된다. 계층  $l$ 에서의 비율은 다음과 같이 정의된다.

$$P^l_{CMSA_S} = \begin{cases} \frac{\text{The number of candidates examined at level } l}{K+N}, & \text{if } K \neq 1. \\ \frac{\text{The number of candidates examined at level } l}{N}, & \text{otherwise} \end{cases} \quad (20)$$

클러스터화된 데이터베이스에서 CMSA<sub>S</sub>는  $K$ 개의

클러스터 중심점들도 함께 조사해야 하기 때문에,  $P_{CMSA_c}^L$ 의 분모는  $K+N$ 이 된다. 예를 들어,  $K=1500$ 일 때 CMSAS는 통계적으로 최상위 계층에서 0.1%의 후보들만을 조사한다. 다시 말해, 나머지 99.9%의 영상 후보들은 하위 계층들에서 모두 제거된다. 한편, 클러스터화되지 않은 데이터베이스에서의 결과와 비교할 때, 클러스터화된 데이터베이스에서의 하위 계층  $P_{CMSA_s}$ 이 상대적으로 작다. 그렇기 때문에, 클러스터화된 데이터베이스에서 매우 빠른 탐색 속도를 얻을 수 있다.

### 3. CMSAM의 성능 평가

$M=5, 10, 15, 20$  등 네 가지 경우에 대해 CMSAM의 성능을 평가하였다 (표 III). 예를 들어  $K=1500$ 이고  $M=5$ 일 때, CMSAM은 ESA보다 평균 25배 정도 빠르다. 한편,  $M$ 이 증가함에 따라 SUR은 감소한다. 이런 현상을 보이는 데는 두 가지 이유가 있는데, 하나는 정렬을 위한 계산량이  $M$ 에 비례하기 때문이다. 다른 하나는  $d_{\min}[M-1]$ 도  $M$ 에 비례해서 증가하기 때문에 각 계층에서의 거리 계산 빈도가 늘기 때문이다. 그러나, CMSAM은 여전히 ESA보다 훨씬 빠른 속도를 보장한다.

표 3. 여러  $M$ 과  $K$ 에 대한 CMSAM의 SUR 비교

Table 3. SUR comparison of CMSAM for various  $M$ 's and  $K$ 's.

		CMSAM			
		$K=1$	$K=500$	$K=1000$	$K=1500$
$M=5$	SUR <sub>AVG</sub>	17.1	22.9	24.2	24.6
	SUR <sub>MAX</sub>	30.3	159	159	127.2
	SUR <sub>MIN</sub>	11.8	11.0	11.8	11.6
$M=10$	SUR <sub>AVG</sub>	13.6	20.2	20.6	20.8
	SUR <sub>MAX</sub>	21.9	127.2	127.2	106
	SUR <sub>MIN</sub>	9.4	9.5	9.5	8.6
$M=15$	SUR <sub>AVG</sub>	11.6	17.7	17.6	18.0
	SUR <sub>MAX</sub>	17.7	106	127.2	106
	SUR <sub>MIN</sub>	8.1	8.8	7.8	7.6
$M=20$	SUR <sub>AVG</sub>	10.2	15.9	15.8	15.8
	SUR <sub>MAX</sub>	14.8	106	106	90.9
	SUR <sub>MIN</sub>	7.2	7.5	7.0	6.9

## IV. 결론

본 논문은 클러스터화된 데이터베이스를 위한 고속 다 해상도 전역 탐색 기법을 제안했다. 데이터베이스는 정해진 수만큼의 클러스터들로 분할되어 있으며, 각 클러스터는 유사한 특징을 갖는 영상들로 구성된다. 제안한 기법은 하나의 최적 정합자는 물론 다수의 최적 정합자들도 매우 빠르고 정확하게 탐색할 수 있다. 하나의 최적 정합자만을 탐색할 경우, ESA보다 평균 40배 정도 빠르며, 기존 기법인 TIA보다 7.5배 가량 빠르다. 다수의 최적 정합자들을 탐색할 경우에도 매우 빠른 탐색 속도를 보장한다. 본 논문에서는 밝기 히스토그램을 영상 특징으로 채택해 사용했지만, 텍스처, 에지, 윤곽선 등 다른 특징들을 이용한 탐색에도 제안한 기법을 적용할 수 있다. 따라서, 제안한 기법은 대용량 멀티미디어 데이터베이스에서의 고속 최적 검색에 적합하다.

## 참고 문헌

- [1] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233-254, 1996.
- [2] M. Flicker, 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.
- [3] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. Shu, "The virage image search engine: An open framework for image management," in *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases*, vol. 2670, pp. 76-87, Feb. 1996.
- [4] J. R. Smith and S. F. Chang, "Exploring image functionalities in www applications-development

- of image/video search and editing engines," in Proc. ICIP, pp. 1-4, Oct. 1997.
- [5] Y. Rui, T. Huang, and S. F. Chang, "Image retrieval: current techniques, promising directions, and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 1, pp. 39-62, 1999.
- [6] S. -K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt, "An intelligent image database system," *IEEE Trans. Software Eng.*, vol. 14, no. 5, 1988.
- [7] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.
- [8] J. Huang, S. R. Kumar, M. Mitra, W. -J. Zhu, and R. Zabih, "Image indexing using color correlograms," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 762-768, 1997.
- [9] X. Wan and C. -C. J. Kuo, "A new approach to image retrieval with hierarchical color clustering," *IEEE Trans. Circ. and Syst. for Video Technol.*, vol. 8, no. 5, pp. 628-643, Sept. 1998.
- [10] S. -C. Pei and C. -M. Cheng, "Extracting color features and dynamic matching for image data-base retrieval," *IEEE Trans. Circ. and Syst. for Video Technol.*, vol. 9, no. 3, pp. 501-512, April 1999.
- [11] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 837-842, Aug. 1996.
- [12] L. M. Kaplan, R. Murenzi, and K. R. Namuduri, "Fast texture database retrieval using extended fractal features," in Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases, vol. 3312, pp. 162-175, Jan. 1998.
- [13] R. Mehrotra and J. Gary, "Similar-shape retrieval in shape data management," *IEEE Computer*, vol. 28, pp. 57-62, Sept. 1995.
- [14] G. C. -H. Chuang and C. -C. J. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Trans. Image Processing*, vol. 5, no. 1, pp. 56-70, Jan. 1996.
- [15] A. P. Berman and L. G. Shapiro, "Efficient image retrieval with multiple distance measures," in Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases, vol. 3022, pp. 12-21, Feb. 1997.
- [16] A. P. Berman and L. G. Shapiro, "Triangle-inequality-based pruning algorithms with triangle tries," in Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases, vol. 3656, pp. 356-365, Jan. 1999.
- [17] J. Shepherd, X. Zhu, and N. Megiddo, "A fast indexing method for multi-resolution nearest neighbor search," in Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases, vol. 3656, pp. 350-355, Jan. 1999.
- [18] Krishnamachari and M. A. Mottaleb, "Hierarchical clustering algorithm for fast image retrieval," in Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases, vol. 3656, pp. 427-435, Jan. 1999.
- [19] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Pacific Groves, CA: Brooks/Cole Publishing Company, 1998.
- [20] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, no. 1, Jan. 1995.
- [21] ISO/IEC JTC1/SC29/WG11/N2466, "Licensing agreement for the MPEG-7 content set," Atlantic City, USA, Oct. 1998.
- [22] Test data sets for MPEG-7 core experiments CT1, CT2, and CT3, in [[ftp://bs.hhi.de/pub/Color DataSet/](ftp://bs.hhi.de/pub/ColorDataSet/)].

---

 저 자 소 개
 

---

## 宋秉哲(正會員)

1972년 11월 8일생. 1994년 2월 한국과학기술원 전기 및 전자공학과 공학사. 1996년 2월 한국과학기술원 전기 및 전자공학과 공학석사. 1996년~현재 한국과학기술원 전기 및 전자공학과 박사과정 재학 중. 주관심분야는 동영상 부호화 및 영상 표현 및 탐색.

## 金明俊(正會員)

1973년 12월 2일생. 1997년 2월 경북대학교 전자공학과 공학사. 1999년 2월 한국과학기술원 전기 및 전자공학과 공학석사. 1999년~현재 한국과학기술원 전기 및 전자공학과 박사과정 재학 중. 주관심분야는 동영상 부호화 및 영상 표현 및 탐색.

## 羅鍾範(正會員)

1953년 5월 23일생. 1975년 2월 서울대학교 전자공학과 공학사. 1977년 2월 한국과학기술원 전기 및 전자공학과 공학석사. 1983년 2월 한국과학기술원 전기 및 전자공학과 공학박사. 1983년 7월~1987년 6월 미국 Columbia 대학교 연구 조교수. 1987년 7월~현재 한국과학기술원 전기 및 전자공학과 교수. 주관심분야는 디지털 영상처리, 동영상 신호처리, 3차원 시각화, 의료영상 시스템 등