

MPEG-7 시각 정보 기술자의 인덱싱 및 결합 알고리즘

(Algorithms for Indexing and Integrating MPEG-7 Visual Descriptors)

송 치 일[†] 낭 종 호^{**}
(Chiill Song) (Jongho Nang)

요약 본 논문에서는 MPEG-7 시각 정보 기술자인 Dominant Color와 Contour Shape 기술자에 대한 새로운 인덱싱 알고리즘을 제안한다. Dominant Color 기술자에서 사용되는 비교 연산 식은 가우스 혼합 모델에 기초하고 있기 때문에 기술자의 각 속성들을 하나의 칼라 히스토그램 형태로 변형시켜서 인덱스로 사용한다. Contour Shape 기술자는 두 단계 형태의 알고리즘을 사용하는데, 첫 번째 단계에서는 글로벌 변수인 Eccentricity와 Circularity를 사용한 대략적인 비교를 통해서 비슷하지 않은 이미지 오브젝트를 배제 시키고 두 번째 단계에서 남겨진 오브젝트들과 질의 오브젝트들간의 Peak 변수를 사용한 비교 연산을 통해 인덱싱을 수행한다. 또한 본 논문은 효율적인 멀티미디어 데이터 검색을 위해서 두 가지의 MPEG-7 시각 정보 기술자 결합 알고리즘을 제안한다. 첫 번째 결합 알고리즘은 가중치를 확률로 변환해서 반영하는 것이고 두 번째는 가중치를 각 비교 연산 결과값의 중요도로 간주하는 방법이다. 실험을 통해서 결과를 분석해 보면 근사화를 통한 인덱스 생성으로 100%의 정확도를 유지 할 수는 없지만 논문에서 제안된 각 기술자의 인덱싱 알고리즘과 기술자들의 결합 알고리즘은 기본 검색 알고리즘과 비교 했을 때 매우 빠른 속도 향상을 보여주었다. 본 논문에서 제안된 알고리즘은 MPEG-7을 사용하는 검색 시스템의 데이터베이스 구축에 효율적으로 사용될 수 있다.

키워드 : 멀티미디어 검색, 고차원 데이터 인덱싱, MPEG-7 시각 정보 기술자

Abstract This paper proposes a new indexing mechanism for MPEG-7 visual descriptors, especially Dominant Color and Contour Shape descriptors, that guarantees an efficient similarity search for the multimedia database whose visual meta-data are represented with MPEG-7. Since the similarity metric used in the Dominant Color descriptor is based on Gaussian mixture model, the descriptor itself could be transform into a color histogram in which the distribution of the color values follows the Gauss distribution. Then, the transformed Dominant Color descriptor (*i.e.*, the color histogram) is indexed in the proposed indexing mechanism. For the indexing of Contour Shape descriptor, we have used a two-pass algorithm. That is, in the first pass, since the similarity of two shapes could be roughly measured with the global parameters such as eccentricity and circularity used in Contour shape descriptor, the dissimilar image objects could be excluded with these global parameters first. Then, the similarities between the query and remaining image objects are measured with the peak parameters of Contour Shape descriptor. This two-pass approach helps to reduce the computational resources to measure the similarity of image objects using Contour Shape descriptor. This paper also proposes two integration schemes of visual descriptors for an efficient retrieval of multimedia database. The one is to use the weight of descriptor as a yardstick to determine the number of selected similar image objects with respect to that descriptor, and the other is to use the weight as the degree of importance of the descriptor in the global similarity measurement. Experimental results show that the proposed indexing and integration schemes produce a remarkable speed-up comparing to the exact similarity search, although there are some losses in the accuracy

[†] 정 회 원 : LG 전자 MC사업본부
chiill@lge.com

^{**} 중 신 회 원 : 서강대학교 컴퓨터학과

jhnang@sogang.ac.kr

논문접수 : 2004년 3월 12일
심사완료 : 2006년 12월 20일

because of the approximated computation in indexing. The proposed schemes could be used to build a multimedia database represented in MPEG-7 that guarantees an efficient retrieval.

Key words : Content based Multimedia Retrieval, High Dimensional Indexing, MPEG-7 Visual Descriptors

1. 서론

컴퓨터 처리 능력의 향상과 인터넷 기술의 급격한 발전은 대용량의 멀티미디어 데이터 처리를 가능하게 했고, 여러 가지 방법으로 멀티미디어 콘텐츠를 서비스 하거나 이용할 수 있게 했다. 다양한 사용자들이 각각의 필요에 따라 멀티미디어 콘텐츠를 저장, 교환 또는 검색함에 있어서 보다 나은 편리성과 정확성을 요구하게 되었고, 이것은 멀티미디어 콘텐츠의 다양한 정보를 기술하고 있는 메타 데이터 기술 표준인 MPEG-7[1-3]을 출현하게 했다. 이런 다양하고 대규모의 멀티미디어 콘텐츠를들 중에서 사용자가 필요로 하는 데이터만을 정확하고 신속하게 찾는다는 것은 매우 어려운 일이며, 이런 어려운 일에 대한 해결은 멀티미디어 데이터 인덱싱 연구를 통해서 풀릴 수 있다. 기존의 멀티미디어 데이터 인덱싱 방법으로는 트리 형태를 이용해서 데이터의 공간을 나누는 다차원 인덱싱 방법[8-10]과 양자화 등의 데이터 압축기법을 사용하는 방법[4] 등이 있다. 그러나 앞에서 언급한 이러한 멀티미디어 데이터 인덱싱 방법들은 MPEG-7 시각 정보 기술자 기반 인덱싱 방법에는 그대로 적용할 수가 없다. 왜냐하면 일반적인 멀티미디어 인덱싱 방법들은 각 차원의 기본 성질이 모두 같은 의미를 내포하는데, MPEG-7 시각 정보 기술자들에 있어서 각 차원의 성질은 그 의미가 같지 않다. 또한 MPEG-7 시각 정보 기술자들은 매우 고차원으로 표현 되는데 기존의 일반적인 멀티미디어 인덱싱 방법들은 이런 고차원을 고려하지 않은 인덱싱 방법이고 또한 고차원을 고려한 인덱싱 방법들도 만족스런 속도와 정확도를 보여주지 못했다. MPEG-7 시각 정보 기술자에는 Dominant Color, Contour Shape, Color Structure, Homogeneous Texture, Edge Histogram, Region Shape이 있으며, Dominant Color와 Contour Shape 기술자는 비트맵 인덱싱[6]를 사용하기 어렵다.

본 논문에서는 MPEG-7 시각 정보 기술자인 Dominant Color와 Contour Shape 기술자에 적합한 새로운 인덱싱 방법과 다양한 검색을 위한 기술자들의 결합 방법을 제안하고, 실험을 통해 그 결과를 분석한다. 먼저 Dominant Color의 인덱싱 방법을 살펴보면 Dominant Color 기술자가 제시하는 비교 연산 식은 가우스 혼합 모델에 기초하므로 의미가 다른 각 속성들을 가우스 혼합 모델에 적용시켜 같은 의미를 내포한 속성 히스토그

램 형태로 변형시킨다. 이렇게 해서 얻어진 속성 히스토그램이 인덱싱의 역할을 하게 되는 것이다. Contour Shape은 두 단계 형태의 알고리즘을 사용하는데, 첫 번째 단계에서는 글로벌 변수 속성들을 그리고 두 번째 단계에서는 비트맵 인덱싱을 사용한 인덱싱을 수행한다. Color Structure, Homogeneous Texture, Edge Histogram, Region Shape 기술자들은 비트맵 인덱싱을 사용해서 인덱싱을 한다. 다양한 검색을 위한 기술자 결합 방법으로는 크게 두 가지가 있는데, 첫 번째는 가중치를 확률로 변환해서 반영하는 것으로서 조절된 가중치는 해당 기술자의 후보 리스트가 최종 후보 리스트에 존재할 확률로 반영이 되는 것이고 두 번째는 가중치를 각 비교 연산 결과값의 중요도로 간주하는 것으로서 조절된 가중치는 해당 기술자의 비교 연산 결과 값에 곱해서 그 값의 중요도를 나타낸다. 실험을 통해서 제안된 인덱싱 알고리즘을 사용한 결과를 분석해 보면 Dominant Color, Contour Shape 기술자의 경우 각각 92%와 84%의 정확도를 유지하면서 검색 속도는 각각 120배와 3배 이상의 향상을 보였고, 나머지 기술자들도 90%의 정확도에 100배 이상의 속도 향상을 나타냈다. 그리고 다양한 검색을 위한 결합 알고리즘의 경우 첫 번째 방법은 95%의 정확도를 두 번째 방법은 82% 정확도를 유지 했으며 속도는 각각 약 50배 이상의 향상을 나타냈다. 본 논문의 제안 알고리즘은 MPEG-7 시각 정보 기술자에 기반한 멀티미디어 데이터 검색 시스템이나 DVL(Digital Video Library) 시스템 등에 유용하게 사용될 것이다.

2. 연구배경

본 장에서는 MPEG-7 시각 정보 기술자[1-3]들의 각 특징 분석과 기존의 멀티미디어 인덱싱 방법에 대한 분석을 한다.

2.1 MPEG-7 시각 정보 기술자의 특징

(1) Dominant Color

이 기술자는 임의의 영역에 대한 주요 색의 집합을 나타내며 최소 한 개에서 최대 8개 사이의 주요 색을 추출한다. 각 구성요소를 살펴보면 Spatial Coherency는 하나의 값을 갖게 되며 각 주요색들의 Spatial Coherency의 가중치 합을 통해서 얻어지는데 주요색과 관련해 각 픽셀들이 얼마나 잘 모여있는가를 의미한다.

다음으로 각각의 주요색들은 Percentage, ColorValue-Index, ColorVariance를 갖게 된다. 먼저 Percentage는 관련된 색을 가지고 있는 픽셀의 Percentage를 나타낸다. 즉, 각 주요색들의 비율을 의미한다. 다음으로, ColorValueIndex는 주요색이 색 양자화에서 선택된 색 공간상의 인덱스 값을 의미한다. 즉, 정의된 색 공간상에서 주요색의 실제 색 값을 의미한다. 마지막으로, ColorVariance는 각 주요색들에 있어서 색 값의 분산을 의미한다. 각종 이미지나 동영상 검색을 위해서 이 기술자가 제시하는 일반적인 비교 연산 식은 식 (1)과 같으며, ColorValueIndex와 ColorVariance를 변수로 사용한 가우스 혼합 모델이라는 방법에 기초하고 있다. 이 식에서 F_1, F_2 는 두 개의 기술자를 나타내고 있고, N_1, N_2 는 F_1, F_2 에서 사용되는 색의 개수, p_{1i}, p_{2i} 는 F_1, F_2 의 i 번째 Dominant Color의 비율을 나타낸다. $c_{xi}^{(l)}, c_{xi}^{(u)}, c_{xi}^{(v)}$ 는 각 x 번째 기술자, i 번째 Dominant Color의 L값, U값, V값을 나타내고, $v_{xi}^{(l)}, v_{xi}^{(u)}, v_{xi}^{(v)}$ 는 x 번째 기술자, i 번째 Dominant Color의 L값, U값, V값의 분산을 나타낸다.

$$D_v(F_1, F_2) = \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} p_{1i} p_{1j} f_{i,j} + \sum_{i=1}^{N_2} \sum_{j=1}^{N_2} p_{2i} p_{2j} f_{i,j} - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2 p_{1i} p_{2j} f_{i,j}$$

$$\text{where, } f_{xi,yj} = \frac{1}{2\pi \sqrt{v_{xi,yj}^{(l)} v_{xi,yj}^{(u)} v_{xi,yj}^{(v)}}} \times \exp\left(-\left(\frac{c_{xi,yj}^{(l)}}{v_{xi,yj}^{(l)}} + \frac{c_{xi,yj}^{(u)}}{v_{xi,yj}^{(u)}} + \frac{c_{xi,yj}^{(v)}}{v_{xi,yj}^{(v)}}\right)/2\right)$$

$$c_{xi,yj}^{(l)} = (c_{xi}^{(l)} - c_{yj}^{(l)})^2, c_{xi,yj}^{(u)} = (c_{xi}^{(u)} - c_{yj}^{(u)})^2, c_{xi,yj}^{(v)} = (c_{xi}^{(v)} - c_{yj}^{(v)})^2$$

$$v_{xi,yj}^{(l)} = (v_{xi}^{(l)} + v_{yj}^{(l)}), v_{xi,yj}^{(u)} = (v_{xi}^{(u)} + v_{yj}^{(u)}), v_{xi,yj}^{(v)} = (v_{xi}^{(v)} + v_{yj}^{(v)}) \quad (1)$$

식 (1) 보면 알 수 있듯이 각 변수의 중요도가 서로 다르게 반영되고 있음을 알 수 있다. 즉, 일반적인 인덱싱 방법을 적용시킬 수 없다는 것이다.

(2) Contour Shape

이 기술자는 이미지나 비디오 연속물에 있어서 2차원 오브젝트나 또는 영역이 닫혀있는 Contour를 나타낸다. 기본적으로 CSS(Curvature Scale Space)에 바탕을 두어서 Contour의 Shape을 표현한다. 구성요소를 살펴보면, CSS에 있어서 Peak의 개수를 의미하는 Number-OfPeaks, Circularity와 Eccentricity를 의미하는 Global-CurvatureVector, 필터링을 거친 후의 Circularity와 Eccentricity를 의미하는 PrototypeCurvatureVector, Peak의 크기가 가장 큰 것의 크기를 의미하는 HighestPeakY, Contour의 x 좌표를 의미하는 xpeak[]와 필터링의 횟수를 의미하는 ypeak[]가 있다. 이 기술자에서 일반적으로 사용되는 비교 연산 식은 식 (2)와 같다. q, r 은 각각 질의와 데이터베이스상의 기술자를 의미하며, $c_q[0]$ 와 $c_r[0]$ 는 각각 질의와 데이터베이스의 Eccentricity 값을 나타내고, $c_q[1]$ 와 $c_r[1]$ 는 각각 질의와 데이터베이스

의 Circularity를 나타낸다. 비교 연산 식을 보면 알 수 있듯이 각 변수의 중요도가 서로 다르게 반영되고 있음을 알 수 있다. 그러므로 일반적인 인덱싱 방법을 사용할 수 없다.

$$D(q,r) = 0.4 \times \frac{|c_q[0] - c_r[0]|}{\text{MAX}(c_q[0], c_r[0])} + 0.3 \times \frac{|c_q[1] - c_r[1]|}{\text{MAX}(c_q[1], c_r[1])} + M_{css} \quad (2)$$

$$M_{css} = \sum_i ((xpeak[i] - xpeak[j])^2 + (ypeak[i] - ypeak[j])^2) + \sum_2 (ypeak[i])^2 \quad (3)$$

(3) 그 밖의 기술자들에 대한 특징

그 밖의 시각 정보 기술자로서 이미지의 색 분포와 색의 공간적 구성을 나타내는 Color Structure 기술자, 주파수 채널의 집합으로부터 에너지와 에너지 편향을 이용해서 영역의 텍스처 정보를 나타내는 Homogeneous Texture 기술자, 로컬 이미지 영역을 5가지 타입의 공간적 분포로 나누어서 표현하는 Edge Histogram 기술자, 한 개의 Region이나 또는 Region들의 집합으로 구성되는 Region Shape 기술자가 있다. 그런데 Homogeneous Texture 기술자와 Edge Histogram 기술자는 각 차원의 성질이 동일하지 않기 때문에 속성 변형이 필요하다.

2.2 기존의 멀티미디어 인덱싱 방법의 분석

고차원의 데이터베이스에 있어서 Similarity Search의 가장 전형적인 방법은 K-nearest neighbor (K-NN) 검색이다. 멀티미디어 데이터베이스에 있어서 실제적인 문제점은 원하는 시간 내에 가장 가까운 K개의 오브젝트를 어떻게 얻느냐인데, 이런 K-NN의 속도를 빠르게 하기 위해서 여러 가지 인덱싱 방법이 연구되고 있다. 먼저, 트리에 기반한 다차원 인덱싱 방법으로서 R*-tree[7], X-tree[8], T-tree[9], SR-tree[10] 등이 있다. 이 방식은 기본적으로 데이터 공간을 나누는 후 각 오브젝트들을 해당되는 공간으로 클러스터링 하는 것으로, 질의가 오게 되면 클러스터링 정보에 의해서 검색 공간을 줄이는 것이다. 차원의 개수가 적은 경우에는 빠른 검색 속도를 기대할 수 있으나 고차원의 경우에는 전체 데이터 공간이 너무 많이 나누어 지기 때문에 나누어진 하나의 데이터 공간에 한 개의 오브젝트도 포함되지 않을 수 있다. 그러므로 인덱스 개수가 오브젝트 개수보다 많아 질 수 있기 때문에 인덱스 검색의 시간이 많이 늘어나게 되므로 오히려 순차검색 보다 더 많은 검색 시간을 요구하게 된다. 이런 고차원의 문제를 해결하기 위해서 제안된 인덱싱 방법이 VA(Vector Approximation) File[4]과 같은 필터링 방법이다. VA-File은 데이터 공간을 셀로 나누어서 데이터 오브젝트 단위로

인덱스를 하는 필터 기반의 인덱싱 방법이다. 필터링을 할 때 모든 오브젝트에 대해서 비교 연산을 수행한 다음에 후보 리스트를 생성하므로 순차검색에 비해 월등한 속도 향상을 나타내지는 못한다. 그런데 MPEG-7 시각 정보 기술자들은 공통적인 특징으로 차원이 매우 높고, Color Structure와 Region Shape의 경우를 제외하고는 각 차원의 성질이 동일하지 않다. 그러므로 이런 기본 특성을 바탕으로 해서 기존의 다양한 멀티미디어 인덱싱 방법을 적용 시킨다면 여러 문제점들이 발생한다. 먼저 트리에 기반한 인덱싱 방법들은 기본적으로 고차원이 아닌 다차원 인덱싱을 목적으로 하고 있기 때문에 MPEG-7 시각정보 기술자의 공통적인 특징인 고차원의 문제를 해결할 수 없다. VA-File 형태의 인덱싱 방법은 기본적으로 고차원의 멀티미디어 데이터에 적합한 인덱싱 방법이지만 각 차원의 성질이 모두 동일하다는 것을 기본 가정으로 하고 있다. 그러므로, 각 차원의 성질이 동일한 Color Structure와 Region Shape을 제외한 나머지 기술자들에 대해서는 그대로 적용시킬 수 없다.

3. MPEG-7 시각 정보 기술자 기반 검색을 위한 기술자 인덱싱 방법 및 결합 방법

3.1 각 기술자의 인덱싱 방법 제안

(1) Dominant Color

Dominant Color는 임의의 영역에서의 주된 색 정보를 기술 하는데, 2.2절에서 언급했듯이 각 속성의 성질이 서로 동일하지 않기 때문에 일반적인 인덱싱 방법을 사용할 수 없다. 그런데 이 기술자가 제시하는 비교 연산 식 식 (1)은 가우스 혼합 모델에 기초 하고 있으므로, 본 논문에서는 이 모델을 이용해서 속성 데이터들을 변형 시킨다. 먼저 Dominant Color 기술자를 나타내기 위해서 사용되는 속성들은 각 색에 대한 ColorValueIndex, ColorVariance, Percentage 가 있고 가우스 분포 식 식 (4)는 이 중 두 가지 속성 σ (ColorVariance) 와 m (ColorValueIndex)을 내포 하고 있다.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (4)$$

즉 가우스 분포 식을 통해서 속성 히스토그램 형태로 변환시킬 수 있는 것이다. 이렇게 각각의 Dominant Color 색들을 속성 히스토그램으로 변형 시킨 다음에는 각각의 속성 히스토그램들을 혼합해야 한다. 예를 들어 Dominant Color의 색이 5개라면 5개의 속성 히스토그램이 나올 것이고 이 5개의 속성 히스토그램을 가우스 혼합 모델에 기초해서 혼합해야 한다. 그림 1을 보면 5개의 Dominant Color 각각에 대한 속성 히스토그램을

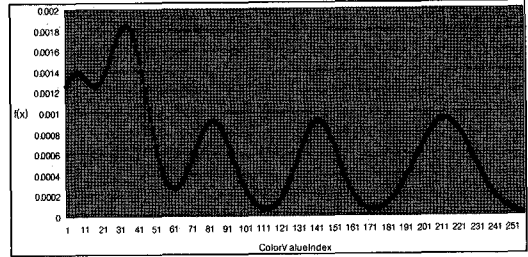


그림 1 5가지 색에 가우스 혼합 모델을 적용한 히스토그램

혼합한 그래프를 보여주고 있다. Dominant Color의 개수가 5개이기 때문에 7, 38, 84, 141, 218의 ColorValueIndex 값을 중심으로 가우스 분포를 띠고 있는 것을 알 수 있다. X축은 0~255 사이의 ColorValueIndex 값을 나타내고 Y 축은 각 색의 가우스 분포 함수 결과 값에 각각의 가중치를 반영한 값이다. 그렇다면 각각의 가중치가 어떻게 반영 되는지를 살펴보자. Dominant Color에 있어서 Percentage의 의미는 관련된 색을 가지고 있는 픽셀의 Percentage를 나타내는데 결과적으로 해당 색이 전체에서 차지하는 비율을 의미하는 것이다. 즉 가우스 혼합 모델에 있어서 가중치의 의미와 동일하다. 그래서 0~31 사이의 값을 갖는 Percentage 속성은 0~1 사이의 값으로 정규화 된 후 해당 가우스 분포에 곱해 저서 가중치 반영이 된다. 지금까지의 방법을 통해서 각각의 서로 다른 성질을 갖는 속성들이 속성 히스토그램 형태로 변형된다. 이렇게 해서 생성된 속성 히스토그램이 인덱스의 역할을 하는 것이다. 이런 식으로 모든 데이터 베이스에 대한 인덱스 작업을 마친 후 질의 오브젝트가 입력으로 오게 되면 그에 맞는 속성 히스토그램 형태의 인덱스를 구성해서 데이터 베이스의 인덱스와 L1-norm 형태의 비교 연산 식을 통해서 후보 리스트를 구성한다. 그리고 구성된 후보 리스트에 대해서만 Dominant Color가 제시하는 비교 연산 식을 통해 최종 결과 리스트를 생성한다.

(2) Contour Shape

Contour Shape의 비교 연산 식 식 (2)를 살펴보면 크게 두 부분으로 나누어서 계산을 하는데 첫 번째 부분이 글로벌 변수들을 다루는 점선으로 된 부분이고 두 번째 부분이 Mcss 부분이다. 그래서, 본 논문에서는 그림 2와 같은 두 단계 형태의 인덱싱 방법을 제안 한다. 그러기 위해서는 먼저 Contour Shape의 인덱스를 구성해야 한다. 인덱스의 구성은 그림 3과 같다.

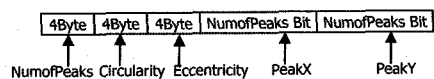


그림 2 Contour Shape의 인덱스 구조

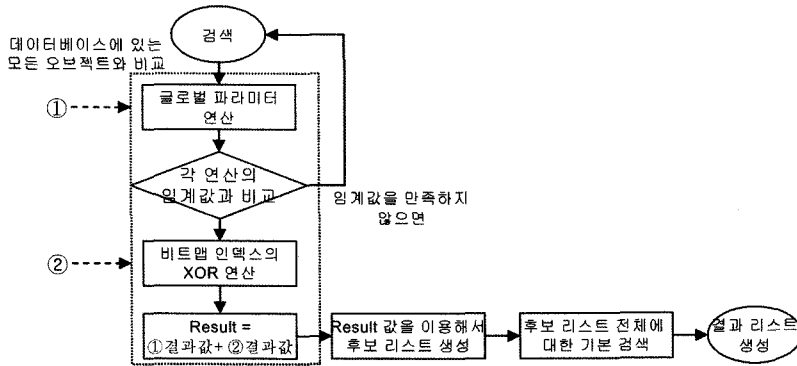


그림 3 Contour Shape의 인덱싱 알고리즘 순서도

먼저, 그림 3의 ①에서는 Contour Shape 기술자의 글로벌 변수인 Circularity와 Eccentricity를 사용한 연산을 한다.

Eccentricity란 이심률로서 이 값이 0이면 원이 되는 것이고 1에 가까워질수록 타원에 가까워지는 의미를 지닌다. 즉 질의로 입력된 오브젝트와 데이터 베이스에 존재하는 오브젝트들간의 전체적인 형태를 원과 타원의 형태를 이용해서 비교하는 식이다. Circularity란 이미지 내 오브젝트의 전체 둘레인 Perimeter의 제곱을 면적인 Area로 나눈 것을 의미한다. 즉 오브젝트가 울퉁불퉁할수록 Circularity 값은 커지게 될 것이다. 다음으로는 ①의 글로벌 변수 연산을 만족한 데이터들에 대해서는 그림 3의 ② 연산을 수행한다. 그리고 ①과 ② 연산의 결과값을 합한 후 이 값을 이용해서 후보 리스트를 생성한다. 후보리스트의 생성이 끝났으면 구성된 후보 리스트에 대해서만 Contour Shape이 제시하는 비교 연산 식을 통해 결과 리스트를 생성한다.

(3) 그 밖의 시각 정보 기술자

Homogeneous Texture 기술자와 Edge Histogram 기술자는 각 차원의 확장을 통해서 비트맵 인덱스를 사용할 수 있고, Color Structure와 Region Shape 기술자는 각 차원의 성질이 동일하고 고차원이기 때문에 2.2 절에 설명한 비트맵 인덱스를 사용해서 인덱싱 할 수 있다.

3.2 기술자들의 결합 방법 제안

MPEG-7 시각 정보 기술자에 기반한 검색을 함에 있어 가중치 조절을 통해 두 개 이상의 기술자들을 결합해서 사용할 수 있다. 그림 4는 기본 검색 알고리즘을 사용한 가중치 결합 방법에 대한 순서도로서 예를 들어 어떤 질의 오브젝트에 대해서 Dominant Color 기술자에 70%, Contour Shape에 30%의 가중치를 설정해서 검색 한다고 하면 다음의 순서를 통해서 결과 리스트를 생성하게 된다. 먼저 질의 오브젝트에 대해서 선택된 기

술자인 Dominant Color와 Contour Shape의 특징 벡터를 Feature 추출기를 통해서 추출한다. 그런 다음 추출된 특징 벡터 들과 메타 데이터의 모든 오브젝트 특징 벡터들을 각 기술자가 정의하고 있는 비교 연산 식을 통해서 비교 연산을 수행한다. 비교 연산을 통해서 계산된 결과 값들은 식 (5)와 같은 정규화 과정을 거쳐서 정규화된 값으로 나오게 된다. x는 각 연산의 결과값을 의미하고 max는 각 연산에서 나올 수 있는 결과값 중 가장 큰 값을 의미한다. 정규화의 과정을 거쳐서 정규화된 값이 나오게 되면 그 값들을 선택된 기술자의 가중치와 곱해서 각각을 더한다. 즉 선택된 기술자들에 대한 Weighted Difference Sum을 구하게 되는 것이다. 그리고 마지막으로 WDS(Weighted Difference Sum)값을 기준으로 가장 작은 WDS 값을 갖는 데이터베이스의 오브젝트들을 결과 리스트로 생성한다.

$$\text{NormalizedValue} = \frac{X}{\text{Max} + X} \quad (5)$$

검색 방법은 질의 오브젝트와 가장 비슷한 K개의 데이터 베이스 오브젝트들을 검색 결과로 나타내는 K-NN (K는 10으로 설정) 검색을 사용한다. 그런데 이런 결합의 방식을 사용해서 검색하는 방법은 여러 개의 기술자를 사용하므로 한 개의 기술자만을 사용해서 검색하는 방법보다 훨씬 더 많은 메타 데이터를 검색해야 한다. 즉 검색할 데이터의 양이 많아지기 때문에 그만큼 검색 시간도 증가하게 된다. 그러므로 본 논문에서는 기술자 결합의 새로운 알고리즘 제안을 통해 효율적인 검색이 이루어지도록 한다. 가중치 반영 방법에 따라 두 가지가 있는데, 첫 번째는 가중치를 확률로 변환해서 반영하는 것으로서 조절된 가중치는 해당 기술자의 후보 리스트가 최종 후보 리스트에 존재할 확률로 반영 되는 것이다. 즉 총 K*10*N 개의 후보 리스트를 생성하고 생성된 후보 리스트에 대해서만 기본 검색 알고리즘을 사용해서 검색하는 방법이다. 사용자가 선택한 기술자의

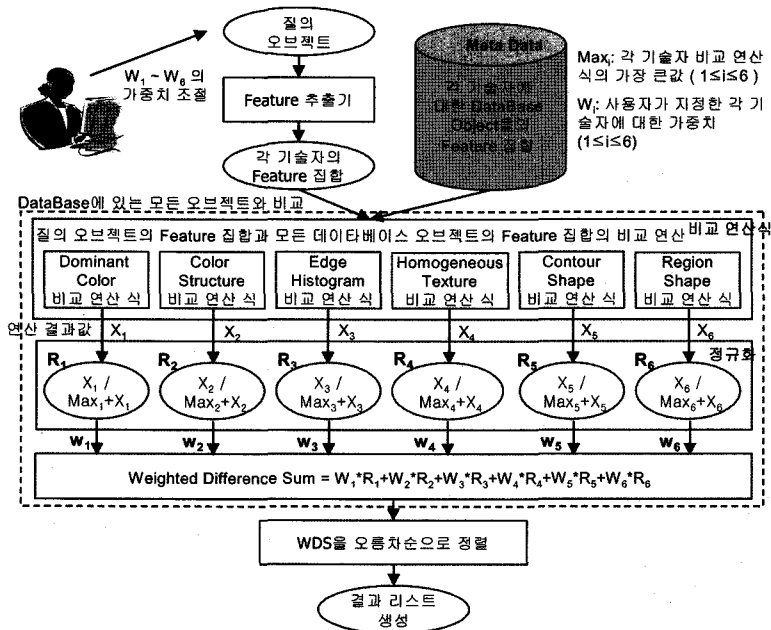


그림 4 가중치를 사용한 기술자 결합 검색 순서도 (기본 검색 알고리즘)

가중치를 최종 결과 리스트에서 해당 기술자의 인덱싱 알고리즘을 통해서 나올 수 있는 결과 리스트의 확률과 연계 시키는 것으로 예를 들어 Homogeneous Texture에 35%, Contour Shape에 65%의 가중치를 통해서 기술자 결합 검색을 할 경우 Homogeneous Texture 인덱싱 알고리즘을 통해서 0.35*(10*10*2)개 즉 70개의 후보리스트를 생성하고 Contour Shape 인덱싱 알고리즘을 통해서 0.65*(10*10*2)개 즉 130개의 후보리스트를 생성하는 것이다. 이렇게 해서 총 200개의 후보 리스트를 생성한 후 기본 검색 알고리즘을 통해서 최종 10개의 결과 리스트를 생성하는 것이다. 두 번째 검색 알고리즘은 가중치를 각 비교 연산 결과값의 중요도로 간주하는 것으로서 조절된 가중치는 해당 기술자의 비교 연산 결과 값에 곱해져서 그 값의 중요도를 나타낸다. 총 K*10*N 개의 후보 리스트를 생성하고 생성된 후보 리스트에 대해서만 기본 검색 알고리즘을 사용해서 검색하는 방법이다. 그리고 선택된 기술자의 가중치는 각 비교 연산 결과값의 중요도인데 예를 들어 Homogeneous Texture에 35%, Contour Shape에 65%의 가중치를 통해서 기술자 결합 검색을 할 경우 Homogeneous Texture 기술자의 비교 연산 결과값에는 0.35를 곱하고 Contour Shape 기술자의 비교 연산 결과값에는 0.65를 곱해서 각각을 더한다. 즉 가중치가 높을수록 해당되는 기술자의 중요도도 높아지게 되는 것이다. 이런 가중치의 의미는 기본 검색 알고리즘에서의 가중치의 의미와 동일하다.

4. 구현 및 실험 결과 분석

4.1 각 기술자의 인덱싱 알고리즘 구현

각 기술자의 전체 검색 시간은 디스크 접근 횟수와 비교 연산 시간의 측정을 통해서 평가했고 검색 성능은 Precision과 Recall의 측정을 통해서 평가했다. 실험 데이터는 12,861개의 Columbia Object Image Library를 사용했다. 검색 시간 중 디스크 접근 횟수에서 사용되는 페이지 크기는 4KB로 가정했다. K-NN 검색에 있어서 K는 10으로 고정했고 인덱싱 검색에서 후보 리스트의 개수는 K의 10배수 즉 K*10개로 정했다. 그리고 성능 평가는 Precision과 Recall의 측정을 통해서 평가했다. 마지막으로 멀티미디어 데이터에 대한 인덱싱의 경우 피드백이나 다른 여러 방법을 사용해서 결과를 향상시킬 수 있고 또한 사용자 역시 90% 정도의 정확도에도 만족을 느낄 수 있기 때문에 꼭 100%의 정확도를 유지시킬 필요는 없다.

4.1.1 Dominant Color 기술자의 인덱싱 알고리즘 분석

(1) 검색 시간

- 디스크 접근 횟수

12861개의 데이터 베이스 오브젝트에 대해서 인덱싱 파일의 크기는 39,508,992 바이트 이고 특징 벡터 집합 파일의 크기는 2,270,452 바이트가 된다. 예를 들어 Dominant Color의 개수를 5라고 가정하면 특징 벡터 집합 파일에서는 한 개의 오브젝트를 나타내기 위해 필요한 바이트 수가 표 1에서 176 바이트이고 인덱스 파

표 1 Dominant Color 개수에 따른 MPEG-7 특징 벡터 집합 크기

Dominant Color 개수	1개	2개	3개	4개	5개	6개	7개	8개
필요한 바이트 수	64	92	120	148	176	204	232	260

일은 Dominant Color의 개수와 관계없이 3072 바이트 (RGB 각각에 대한 float형의 256빈으로 구성)가 필요하다. 즉 한 개의 오브젝트를 표현하기 위한 메모리 양이 인덱스 파일이 MPEG-7 특징 벡터 집합 파일보다 훨씬 더 크다는 것을 알 수 있다. 그러므로 디스크 접근 횟수도 인덱스 검색이 MPEG-7 기본 검색보다 더 많다.

• 비교 연산 시간

그림 5에서 인덱스 검색이 MPEG-7 기본검색 보다 약 143배 더 빠르다는 것을 알 수 있다. MPEG-7 기본 검색에서의 비교 연산 식은 가우스 혼합 모델을 이용하므로 연산식이 매우 복잡하고 시간이 오래 걸리지만 인덱스 검색의 경우 Euclidian Distance의 비교 연산 식을 사용하기 때문에 매우 빠른 속도 향상을 나타낸다.

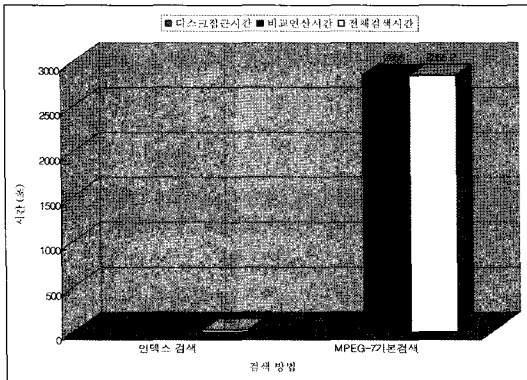


그림 5 Dominant Color에서 전체 검색 시간 비교

• 전체 검색 시간

그림 5에서 인덱스 검색이 MPEG-7 기본 검색 보다 전체 검색 시간에 있어 약 123배 빠르다는 것을 알 수 있다. 디스크 접근 횟수는 인덱스 검색이 더 많지만 MPEG-7 기본 검색은 전체 검색 시간에서 디스크 접근 시간이 차지하는 비중이 매우 작다. 그러므로 인덱스 검색이 MPEG-7 기본 검색 보다 디스크 접근 속도는 느리지만 비교연산 속도에 있어서 훨씬 빠르기 때문에 전체 검색 시간이 빨라지게 된다. 그림 6은 오브젝트 개수에 따른 전체 검색 시간의 변화를 나타내고 있는데, 인덱스 검색은 오브젝트 개수의 증가에도 검색 시간이 매우 완만하게 증가하지만 MPEG-7 기본 검색의 경우 오브젝트 개수가 증가함에 따라 전체 검색 시간이 급격하게 증가하는 것을 알 수 있다.

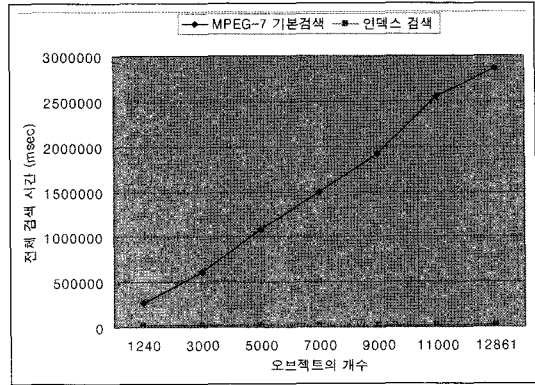


그림 6 Dominant Color에서 오브젝트 개수의 변화에 따른 전체 검색 시간 변화

(2) 검색 성능 및 결과 분석

50번의 검색을 통해서 Precision과 Recall의 평균을 측정한 결과 각 92%의 정확도를 유지하면서 약 120배 이상의 속도 향상을 나타냈다. 그런데 인덱스 파일 생성에 있어서 정보의 손실이 발생하기 때문에 100%의 정확도는 유지할 수 없다.

4.1.2 Contour Shape 기술자의 인덱싱 알고리즘 분석

(1) 검색 시간

• 디스크 접근 횟수

먼저 12861개의 데이터 베이스 오브젝트에 대해서 인덱스 파일의 크기는 256,593 바이트이고 특징 벡터 집합 파일의 크기는 3,178,692 바이트이다. 인덱스 검색의 총 디스크 접근 횟수는 256번이고 MPEG-7 기본 검색은 776번이다. 인덱스 검색이 약 1/3번 디스크 접근을 한다.

• 비교 연산 시간

그림 7을 보면 인덱스 검색이 MPEG-7 기본검색 보다 약 2.3배 더 빠르다는 것을 알 수 있다. 인덱스 검색은 비트 연산을 통해서 비교 연산을 수행하므로 MPEG-7 기본 검색의 비교 연산 방법 보다 더 빠른 속도를 나타낸다.

• 전체 검색 시간

그림 7에서 인덱스 검색의 전체 검색 시간은 198msec (디스크접근시간: 12msec, 비교연산시간: 186msec)이고 MPEG-7 기본 검색의 전체 검색 시간은 605msec (디스크접근시간: 159msec, 비교연산시간: 446msec) 이다. 인덱스 검색 방법이 데이터 접근 시간에 있어서 약 13배, 비교연산속도에 있어서 약 2.3배 정도의 빠른 속도

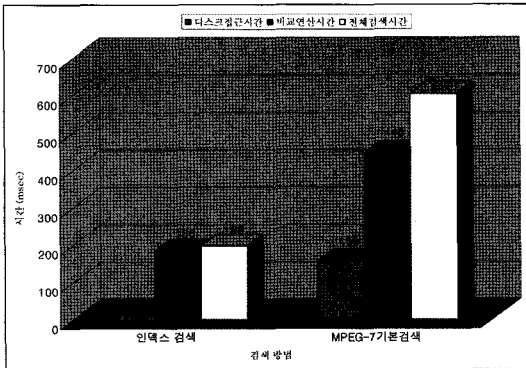


그림 7 Contour Shape 에서 전체 검색 시간 비교

를 나타냈다. 그리고 전체 검색 속도는 인덱스 검색이 MPEG-7 기본 검색 보다 약 3배 빠르다는 것을 알 수 있다. 또한, 인덱스 검색은 오브젝트의 개수가 증가해도 검색 시간이 완만하게 증가하지만 MPEG-7 기본 검색의 경우 오브젝트 개수가 증가함에 따라 전체 검색 시간이 급격하게 증가하는 것을 알 수 있다.

(2) 검색 성능 및 결과 분석

Dominant Color와 같은 방식으로 성능 평가를 했고, 그런 결과 Precision, Recall이 각각 84%로 측정됐다. 그런데 100%의 정확도를 유지할 수 없는 것은 식 (2)의 비교 연산 식 M에 있어서 앞부분의 글로벌 특징 벡터의 계산과 식 (3)의 Mcss 부분의 계산에 있어서 각 결과 값의 정규화 문제가 발생 할 수 있고, 다음으로 Contour Shape 기술자의 차원이 이론상으로는 128차원까지 나올 수 있지만 실제적으로는 20차원 정도에서 결정 되므로 차원이 매우 높다고 할 수 없다. 그리고 오브젝트 추출의 어려움으로 인해서 12861개의 데이터 베이스 오브젝트 중 1240개 만이 실제 데이터이고 나머지 11621개는 랜덤 하게 생성한 가상 데이터이다. 이런 이유들로 해서 전체적인 정확도가 떨어지는 것이다. 그럼에도 불구하고 Contour Shape 기술자의 인덱싱 알고리즘을 사용한 검색은 전체적으로 84%의 정확도를 유지 하면서 검색 속도를 약 3배 이상 향상시켰다.

4.2 가중치 조절을 통한 기술자들의 결합 알고리즘 결과 분석

(1) 전체 검색 시간

결합 알고리즘 1과 결합 알고리즘 2를 분석해 보면 비교 연산 횟수가 같다는 것을 알 수 있다. 디스크 접근 시간이 전체 검색 시간에서 매우 작은 부분을 차지하는 것을 고려하면 이 두 알고리즘의 전체 검색 시간은 거의 같다. 그리고 실험을 통해서도 두 검색 알고리즘의 전체 검색 시간은 거의 비슷하게 산출됐다. 기술자 결합의 기본 검색 알고리즘은 각 기술자의 기본 검색 알고

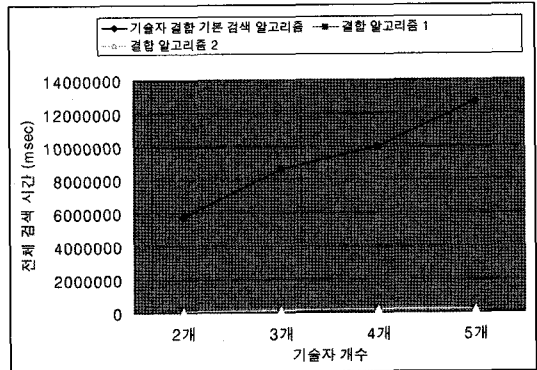


그림 8 기술자 개수 변화에 따른 전체 검색 시간 비교

리즘을 사용해서 구현한 것이기 때문에 기술자 결합 검색의 전체 검색 시간은 각 기술자의 기본 검색 알고리즘의 검색 속도와 비례 관계가 있다. 그리고 결합 알고리즘 1과 2 모두 생성되는 후보 리스트의 개수는 $K*10*N$ 으로 설정하기 때문에 전체 검색 시간은 결합되는 기술자의 개수가 증가함에 따라 점차 늘어날 것이다. 그러나 기술자 개수가 5인 경우라도 그림 8을 보게 되면 결합 알고리즘 1이나 결합 알고리즘 2와 같은 제안된 기술자 결합 검색 알고리즘이 기술자 결합 기본 검색 알고리즘보다 50배 이상 더 빠른 속도를 나타내고 있다.

(2) 검색 성능 및 결과 분석

기술자 개수의 증가에 따른 결합 알고리즘 1과 결합 알고리즘 2의 정확도를 분석해 보자. 결합되는 기술자의 개수가 증가 하더라도 정확도에는 영향을 끼치지 않는다. 이는 기술자 개수가 증가함에 따라서 생성되는 후보 리스트의 개수를 증가 시켰기 때문이다. 또한, 결합 알고리즘 1 보다 결합 알고리즘 2가 더 높은 정확도를 보이는데, 이것은 결합 알고리즘 2에서의 가중치 역할이 기본 검색 알고리즘에서의 가중치와 동일한 의미로 사용되기 때문에 더 좋은 정확도를 나타낸 것이다. 기술자 결합에 있어서 생성되는 후보 리스트 개수는 $K*10*N$ 으로 설정하는데 결합되는 기술자 개수가 증가 할수록 각 기술자의 비교 연산 결과값 정규화 문제로 인한 오차가 누적되므로 이를 보완하기 위해서 후보 리스트 개수를 증가 시킨 것이다. 생성되는 후보 리스트의 개수 증가는 전체 검색 시간을 약간 느려지게 할 수 있지만 N이 6인 경우 즉 결합되는 기술자의 개수가 6개인 경우에도 기본 검색 알고리즘에 비해서 50배 이상의 속도를 나타내고 있다. 전체적인 성능을 살펴보게 되면 결합 알고리즘 2는 약 95% 정도의 정확도를 유지하면서 기본 검색 알고리즘에 비해 약 50배 이상의 속도 향상을 나타냈고, 결합 알고리즘 1은 약 82%의 정확도에 역시 50배 이상의 속도 향상을 나타냈다.

4.3 관련 연구와의 비교

고차원 데이터를 위한 인덱싱 구조로 제안된 것이 IGrid 인덱싱 알고리즘이다. 이 알고리즘은 동일한 개수의 오브젝트들로 구성되게 각 차원의 영역을 나누고 비교되는 두 개의 오브젝트에 있어 각 차원 중 같은 영역이 존재하는 차원에 대해서만 비교 연산을 수행한다. 예를 들어, 두 개의 오브젝트 $X = \{x_1, \dots, x_d\}$ 와 $Y = \{y_1, \dots, y_d\}$ 가 있다고 하자. 오브젝트 X, Y 모두 d 차원으로 구성되어 있다. 차원 i 에서 x_i 와 y_i 가 같은 영역에 속해 있다고 하자. 그러면 x_i 와 y_i 는 차원 i 에 대해서 Proximity라고 말한다. 그리고 이런 각 차원의 Proximity들을 모아놓은 것이 Proximity 집합이 된다. 두 오브젝트의 실제적인 비교 연산은 Proximity 집합에 대해서만 수행한다. 그러나 IGrid 인덱싱의 방법은 그림 9와 같이 c 와 d 는 비슷한 오브젝트이지만 서로 다른 영역으로 나누어 질 수 있기 때문에 차원이 높아질수록 에러 값이 누적되어 정확도에 많은 영향을 끼칠 것이다.



그림 9 기술자 개수 변화에 따른 전체 검색 시간 비교

그리고 IGrid 인덱싱의 방법은 데이터들이 밀집해 있는 경우 매우 효율적으로 사용되는데 MPEG-7 시각 정보 기술자들의 데이터들은 값들이 매우 편향되어 있다. 그러므로 효율적인 검색을 기대 할 수 없다. IGrid 인덱싱 역시 각 차원의 성질이 모두 동일하다는 것을 기본 가정으로 하고 있기 때문에 각 차원의 성질이 동일하지 않은 Color Structure와 Region Shape을 제외한 나머지 기술자들에 대해서는 그대로 적용 시킬 수 없다.

5. 결론 및 앞으로의 연구 방향

일반적인 멀티미디어 인덱싱 방법들은 각 차원의 기본 성질이 모두 같은 의미를 내포하는데 MPEG-7 시각 정보 기술자들에 있어서 각 차원의 성질은 그 의미가 같지 않다. 또한 MPEG-7 시각 정보 기술자들은 매우 고차원으로 표현 되는데 기존의 일반적인 멀티미디어 인덱싱 방법들은 이런 고차원을 고려하지 않은 인덱싱 방법이고 또한 고차원을 고려한 인덱싱 방법들도 만족스런 속도와 정확도를 보여주지 못했다. 그래서 MPEG-7 시각 정보 기술자들에 대한 새로운 인덱싱 알고리즘이 필요하다.

본 논문에서는 Dominant Color와 Contour Shape 기술자에 적합한 새로운 인덱싱 방법을 제안 했고 다양한 검색을 위한 기술자들의 결합 방법을 제안했다. 그리고

실험을 통해서 Dominant Color 기술자는 제안된 인덱싱 알고리즘을 사용했을 때 90%의 정확도를 유지하면서 120배 이상의 속도 향상을 나타냈고 Contour Shape 기술자는 82%의 정확도를 유지하면서 3배 이상의 속도 향상을 나타냈다. 다양한 검색을 위한 기술자들의 결합 알고리즘은 크게 두 가지로 제안 했는데, 각각의 알고리즘은 실험을 통해서 기본 검색 알고리즘에 비해 50배 이상의 속도 향상을 나타냈고 정확도는 결합 알고리즘 1이 82%의 정확도를, 결합 알고리즘 2가 95%의 정확도를 나타냈다.

기술자 결합 알고리즘 1, 2 에 있어서 생성되는 후보 리스트의 개수는 $K*10*N$ 으로 설정했는데 이는 기술자 개수인 N 이 증가 할수록 각 기술자의 비교 연산 결과값 정규화 문제로 인한 오차 누적을 보완할 목적 이었다. 그러나 여전히 각 기술자의 비교 연산 결과값들을 정규화 한 값들이 한 쪽으로 편향되는 현상은 해결되지 않았다. 또한 Contour Shape의 경우 현재 84% 정도의 정확도를 나타내고 있는데 보다 정확하고 자세한 연산식을 사용할 수 있는 알고리즘의 제안을 통해 정확도를 높여야 할 것이다.

참고 문헌

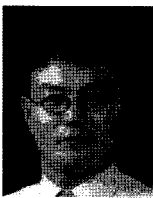
- [1] B.S. Manjunath, Philippe Salembier and Thomas Sikora, *Introduction to MPEG-7 Multimedia Content Description Interface*, JOOHN WILEY & SONS, 2002.
- [2] ISO/IEC JTC 1/SC 29/WG 11, *Information Technology - Multimedia Content Description Interface - Part 3: Visual*, 2001.
- [3] ISO/IEC JTC1/ SC29/WG11/ N4031, *Overview of the MPEG-7 Standard (version 5.0)*, March 2001.
- [4] R. Weber, H.J. Schek, and S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces," *Proceedings of ICVLDB*, pp.194-205, 1998.
- [5] G.H. Cha, X. Zhu, D. Petkovic, and C.W. Chung, "An Efficient Indexing Method for Nearest Neighbor Searches in High-Dimensional Image Databases," *IEEE Transaction on Multimedia*, Vol. 4, No. 1, pp. 76-87, 2002.
- [6] Jingung Jeong and Jongho Nang, "An Efficient Bitmap Indexing Method for Similarity Search in High Dimensional Database," *ICME2004* 제출중.
- [7] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," *Proceedings of ACM SIGMOD Conference*, pp. 322-331, 1990.
- [8] S. Berchtold, D.A. Keim, and H.P. Kriegel, "The X-tree: An Index Structure for High-Dimensional Data," *Proceedings of 22nd International Conference*

- Very Large Databases*, pp. 28-39, 1996.
- [9] H. Lu, Y. Yeung Ng, and Z. Tian, "T-tree or B-tree : Main Memory Database Index Structure," *Proceedings of 12th Australasian Database Conference*, pp. 263-270, 2001.
- [10] N. Katayama, and S. Satoh, "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," *Proceedings of ACM SIGMOD Conference*, pp. 369-380, 1997.
- [11] C.C. Aggarwal, and P.S. Yu, "The IGrid Index: Reversing the Dimensionality Curse for Similarity Indexing in High Dimensional Space," *Proceedings of ACM SIGKDD International Conference*, pp. 119-129, 2000.
- [12] E. Tuncel, H. Ferhatosmanoglu, and K. Rose, "VQ-Index: An Index Structure for Similarity Searching in Multimedia Databases," *Proceedings of ACM International Conference on Multimedia*, pp. 543-552, 2002.
- [13] K.V.R. Kanth, D. Agrawal, and A. Singh, "Dimensionality Reduction for Similarity Searching in Dynamic Databases," *Proceedings of ACM SIGMOD Conference*, pp. 166-176, 1998.



송 치 일

2002년 서강대학교 컴퓨터학과 학사. 2004년 서강대학교 컴퓨터학과 석사. 2004년~현재 LG 전자 MC 사업본부



남 중 호

1986년 2월 서강대학교 전자계산학과 졸업(학사). 1988년 2월 한국과학기술원 전산과 졸업(석사). 1992년 2월 한국과학기술원 전산과 졸업(박사). 1992년 3월~1992년 8월 한국과학기술원 정보전자 연구소(연구원). 1992년 9월~1993년 8월 일본 Fujitsu 연구소(방문연구원). 1993년 9월~현재 서강대학교 컴퓨터학과 교수. 관심분야는 멀티미디어 시스템, 동영상 검색, 동영상 분석, 병렬/분산 처리, 인터넷 컴퓨팅