

# 효율적인 멀티미디어 검색을 위한 MPEG-7 시각 정보 기술자의 인덱싱 및 결합 알고리즘

송치일 김재영<sup>o</sup> 정진국 낭종호  
서강대학교 컴퓨터학과  
young0kr@mlneptune.sogang.ac.kr<sup>o</sup>

## An Indexing and Integration Schemes of MPEG-7 Visual Descriptors for Efficient Multimedia Retrievals

Chiil Song, Jaeyoung Kim<sup>o</sup>, Jinguik Jung, Jongho Nang  
Department of Computer Science, Sogang University

### 요 약

최근 멀티미디어 정보를 기술하기 위한 표준인 MPEG-7이 제안되어 이미지/동영상 검색 시스템과 같은 응용분야에서 사용되기 시작하였다. 그러나 MPEG-7 시각 정보 기술자들은 대부분 고차원으로 표현되고 기술자들이 가지는 각 속성들의 성질이 서로 동일하지 않기 때문에 기존의 인덱싱 방법으로는 효율적인 검색을 할 수 없다. 본 논문에서는 MPEG-7 시각 정보 기술자중에서 많이 사용되는 Dominant Color 기술자와 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 2 단계 형태의 알고리즘을 사용한다. 각 단계는 글로벌 파라미터 속성과 비트맵 인덱스를 사용한 인덱싱이 적용된다. 제안된 인덱싱 방법을 사용했을 때 Dominant Color의 경우 90%의 정확도에 120 배 이상의 속도 향상을 나타냈고, Contour Shape의 경우 82%의 정확도에 3배 이상의 속도 향상을 나타냈다.

### 1. 서 론

컴퓨터 처리 능력의 향상과 인터넷 기술의 급격한 발전으로 대용량의 멀티미디어 데이터처리를 가능하게 됐고, 여러 가지 방법으로 멀티미디어 콘텐츠를 서비스 하거나 이용할 수 있게 됐다. 최근 멀티미디어 정보를 기술하기 위한 표준인 MPEG-7[1]이 제안되어 이미지/동영상 검색 시스템과 같은 응용분야에서 사용되기 시작하였다. 특히 동영상 검색 시스템에서 원하는 데이터만을 정확하고 신속하게 찾는다는 것은 매우 어려운 일이며, 이러한 문제의 해결을 위해 새로운 인덱싱 방법이 요구된다.

멀티미디어 콘텐츠에 대한 검색의 속도를 빠르게 하기 위해 다차원 인덱싱 방법[2,3]과 VA-File[4] 등과 같은 고차원 인덱싱 방법들이 연구되고 있다. 다차원 인덱싱은 차원을 줄이거나 압축 기법을 이용하는데, 고차원 데이터 공간(100개 이상)을 고려하지 않았기 때문에 고차원에서의 검색 속도가 느려진다는 문제점이 있다. 그리고 고차원 인덱싱의 경우에는 계산량이 복잡하다는 문제점을 가지고 있다. 이러한 문제점들

을 해결하기 위해 비트맵 인덱스를 사용한 인덱싱 방법[5]이 제안 됐다. 비트맵 인덱스는 각 오브젝트들이 가지고 있는 속성들을 이진 형태로 나타낸 인덱스로서, 어떤 오브젝트에 대한 인덱스의 각 비트는 이 오브젝트가 특정 조건을 만족시키는 지를 나타낸 것이다. 하지만 MPEG-7의 Dominant Color와 Contour Shape 시각 정보 기술자는 각 속성들의 의미가 서로 달라서 비트맵 인덱스의 구성이 어렵고, 또한 구성을 하더라도 정확도가 많이 떨어지기 때문에 새로운 인덱싱 방법이 필요하다. 본 논문에서는 MPEG-7 시각 정보 기술자들에 적합한 새로운 인덱싱 방법을 제안한다.

### 2. MPEG-7 시각 정보 기술자의 인덱싱 방법

Dominant Color 기술자는 임의의 영역에서의 주된 색 정보를 기술한다. Dominant Color 기술자로 멀티미디어 데이터를 표현하기 위해서 각 색에 대한 Color Value Index, Color Variance, Percentage 등의 속성들이 사용된다. 그런데 이러한 속성들의 성질은 서로 동일하지 않기 때문에, 각 속성들이 동일한 성질임을 기본으로 가정하는 일반적인 인덱싱 방법에 그대로 적용할 수 없다. 따라서 인덱싱을 위해서는 각 속성들의 성질을 동일하게 변경하는 작업이 필요하다. Dominant Color 기술자의 비교 연산 식은 가우스 혼합 모델에 기초하고 있다. 그러므로 가우스 혼합 모델을 이용하면 속성 데이터

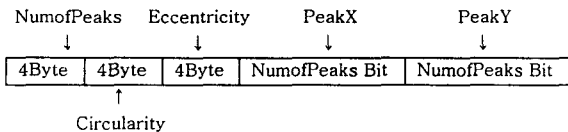
· 본 연구는 한국과학재단에서 지원하는 특장기초연구사업으로 수행하였음 (과제번호 : R01-2002-000-00141-0)

들의 형태를 변형할 수 있다. 아래의 <식 1>은 가우스 분포 식이다. 가우스 분포 식은 두 가지의 속성  $\sigma^2$ (Color Variance)와  $m$ (Color Value Index)을 내포하고 있다. 즉 가우스 분포 식을 통하면 Dominant Color 기술자를 속성 히스토그램 형태로 변형이 가능하다.

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

각각의 Dominant Color를 속성 히스토그램으로 변형시킨 다음에는 각각의 속성 히스토그램들을 혼합해야 한다. 가우스 혼합 모델에 있어서 각 가우스 분포를 혼합할 때는 가중치를 고려하게 된다. 여기서 가중치는 전체 가우스 혼합 분포에 있어서 해당 가우스 분포가 발생할 확률로서 정해진다. Dominant Color에 있어서 Percentage의 의미는 관련된 색을 가지고 있는 픽셀의 Percentage를 의미하는데, 결과적으로 해당 색이 전체에서 차지하는 비율을 나타낸다. 즉, 가우스 혼합 모델에 있어서 가중치의 의미와 동일하다. 이와 같은 방법을 통하여 각각 서로 다른 성질을 갖는 속성들이 속성 히스토그램 형태로 변형되며, 생성된 속성 히스토그램은 인덱스의 역할을 한다. 이렇게 모든 데이터베이스에 대한 인덱스 작업을 마친 후, 질의 오브젝트가 입력으로 오게 되면 그에 맞는 속성 히스토그램 형태의 인덱스를 구성한다. 그리고 데이터베이스의 인덱스와 L1-norm 형태의 비교 연산 식을 통해 후보 리스트를 구성한다. 이와 같이 구성된 후보 리스트에 대해서만 Dominant Color가 제시하는 비교 연산 식을 적용하여 최종 결과 리스트를 생성한다.

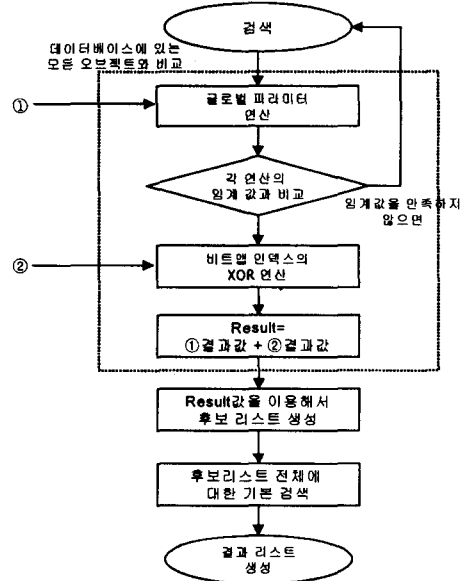
Contour Shape 기술자는 CSS(Curvature Scale Space)에 기초하여 Contour Shape을 표현한다. CSS의 각 구성요소를 살펴보면, peak의 개수를 의미하는 numberOfPeaks, Circularity와 Eccentricity를 의미하는 GlobalCurvature-Vector, 필터링을 거친 후의 Circularity와 Eccentricity를 의미하는 PrototypeCurvatureVector, 가장 큰 peak을 의미하는 HighestPeakY, 그리고 contour의 x 좌표를 의미하는 PeakX[]와 필터링의 횟수를 의미하는 PeakY[]가 있다. Contour Shape의 경우 서로 다른 중요도의 속성들이 비교 연산 식에서 사용되므로, <그림 2>와 같은 2 단계 형태의 인덱싱 방법을 제안한다. 인덱싱을 위해 우선 <그림 1>과 같이 Contour Shape의 인덱스를 구성한다.



<그림 1> Contour Shape의 인덱스 구조

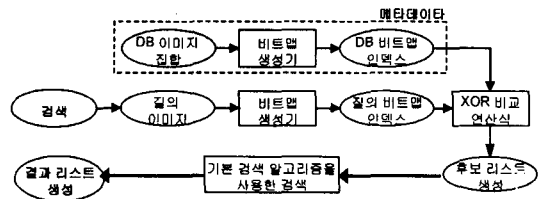
그리고 <그림 2>의 순서도와 같이 인덱싱이 처리된다. ①에서는 글로벌 파라미터인 Circularity와 Eccentricity를 사용한 연산을 한다. 다음으로는 ①의 글로벌 파라미터 연산을 만족하는 데이터들에 대해 ②연산을 수행한다. ②연산은 비트맵 인덱스에 대한 비트 부분의 XOR을 수행하는 것인데, 결과로

나온 1의 개수를 모두 더하는 것으로서 결과 값을 얻는다. 그리고 ①과 ②연산의 결과 값을 합한 후, 이 값을 이용해서 후보 리스트를 생성한다. 이렇게 구성된 후보 리스트에 대해서만 Contour Shape이 제시하는 비교 연산 식을 통해 결과 리스트를 생성한다.



<그림 2> Contour Shape의 인덱싱 알고리즘 순서도

이외의 시각 정보 기술자인 Color Structure, Homogeneous Texture, Edge Histogram, Region Shape 기술자는 각 변수의 중요도가 같은 L1-norm 형태의 비교 연산 식을 가지고 있다. 따라서 <그림 3>과 같이 각 변수에 대한 비트맵 인덱스[5]를 구성한 후, Exclusive OR 연산을 통해 후보 리스트를 생성한다. 그리고 생성된 후보 리스트에 대해서만 Region Shape이 제시하는 비교 연산 식을 사용하여 검색한다.



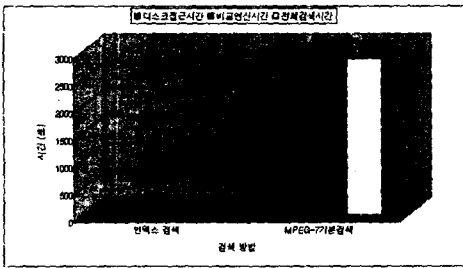
<그림 3> 비트맵 인덱싱 알고리즘 순서도

### 3. 실험 및 분석

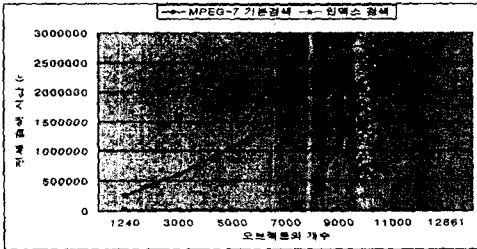
각 기술자의 전체 검색 시간은 디스크 접근 횟수와 비교 연산 시간의 측정을 통해서 평가되었고 검색 성능은 Precision과 Recall의 측정을 통해서 평가되었다. 실험 데이터는 12,861개의 Columbia Object Image Library를 사용했다. 검색 시간 중 디스크 접근 횟수에서 사용되는 페이지 크기는

4KByte로 가정했다. K-NN(Nearest Neighbor) 검색에 있어서 K값은 10으로 고정했고, 인덱싱 검색에서 후보 리스트의 개수는 K의 10배수인  $K*10$ 개로 정했다.

<그림 4>는 Dominant Color에 제안된 인덱싱 알고리즘을 사용했을 때의 전체 검색 시간 비교 그래프이다. 결과로 90%의 정확도를 유지하면서 120배 이상의 속도 향상을 보여주었다. <그림 5>는 오브젝트 개수에 따른 전체 검색 시간의 변화를 나타내고 있다. 여기서 인덱싱 검색은 오브젝트 개수가 증가해도 검색 시간이 매우 완만하게 증가했지만, MPEG-7 기본 검색의 경우에는 오브젝트 개수가 증가함에 따라 전체 검색 시간이 급격하게 증가했다.

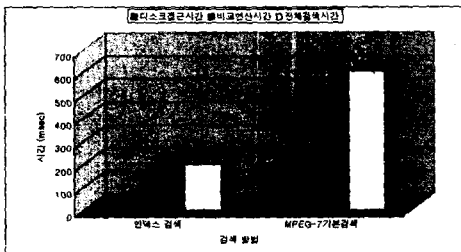


<그림 4> Dominant Color에서 전체 검색 시간 비교

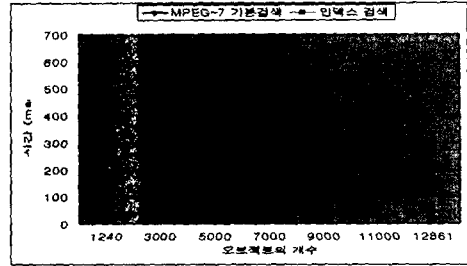


<그림 5> Dominant Color에서 오브젝트 개수의 변화에 따른 전체 검색 시간 변화

<그림 6>은 Contour Shape에 제안된 인덱싱 방법이 사용되었을 때의 전체 검색 시간을 나타낸 그래프이다. Contour Shape의 경우 82%의 정확도를 유지하면서 3배 이상의 속도 향상을 나타냈다. 그리고 <그림 7>에서 볼 수 있듯이 Dominant Color와 마찬가지로 오브젝트의 개수가 증가해도 인덱싱 검색의 검색 시간이 완만하게 증가했다.



<그림 6> Contour Shape에서 전체 검색 시간 비교



<그림 7> Contour Shape에서 오브젝트 개수의 변화에 따른 전체 검색 시간 변화

나머지 MPEG-7 시각 정보 기술자들도 비트맵 인덱스를 사용한 인덱싱 알고리즘[5]을 적용시킨 결과 각각 90%의 정확도를 유지하면서 100배 이상의 속도 향상을 나타냈다.

#### 4. 결론 및 향후 연구 방향

본 논문에서는 Dominant Color와 Contour Shape 기술자에 적합한 새로운 인덱싱 방법을 제안했고, 다양한 검색을 위한 기술자들의 결합 방법을 제안했다. 기술자 결합 알고리즘에 있어서 생성되는 후보 리스트의 개수는  $K*10*N$ 으로 설정했는데 이는 기술자 개수인 N값이 증가 할수록 각 기술자의 비교 연산 결과 값 정규화 문제로 인한 오차 누적을 보완할 목적이었다. 그러나 여전히 각 기술자의 비교 연산 결과 값들을 정규화한 값들이 한 쪽으로 편향되는 현상은 해결되지 않았다. 또한 Contour Shape의 경우 현재 84% 정도의 정확도를 나타내고 있는데, 이는 보다 정확하고 자세한 연산 식을 사용하는 알고리즘의 제안을 통해 정확도를 높여야 할 것이다.

#### 5. 참고문헌

- [1] B.S. Manjunath, Philippe Salembier and Thomas Sikora, *Introduction to MPEG-7 Multimedia Content Description Interface*, JOHN WILEY & SONS, 2002.
- [2] H. Lu, Y. Yeung Ng, and Z. Tian, " T-tree or B-tree : Main Memory Database Index Structure," *Proceedings of 12<sup>th</sup> Australasian Database Conference*, pp. 263-270, 2001.
- [3] 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.
- [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] Jinguik Jeong and Jongho Nang, " An Efficient Bitmap Indexing Method for Similarity Search in High Dimensional Database," ICME2004 (제출중).