멀티미디어 정보관리 데이터베이스 시스템에서 자연어를 사용한 정보 검색

이현창* · 배상현**

Information Retrieval Using Natural Language for Multimedia Information Management Database System

Hyun-chang Lee* · Sang-hyun Bae**

이 논문은 2003년도 조선대학교 학술연구비의 지원을 받아 연구되었음

요 약

오늘날 사용자가 요구하는 데이터 타입은 주로 멀티미디어 데이터 타입들이다. 이들 멀티미디어 데이터 타입의 특성은 기존의 데이터에 비하여 데이터의 크기가 크다는데 있다. 멀티미디어 데이터는 크기가 크기 때문에 멀티미디어 데이터 탐색 연산시 한번에 여러 데이터를 주기억 장치에 가져올 수 없으며, 이것은 많은 입출력 발생과 멀티미디어 데이터 시스템의 성능을 저하시키는 요인이 된다. 그러므로 본 논문에서는 보다 신속한 멀티미디어 데이터 접근을 이루기 위해 인덱스 방법에 관해 살펴보며, 이 기술을 이용하여 멀티미디어 데이터 접근을 많이 요구하는 응용프로그램에 적절하게 대처할 수 있으며, 사용자는 자연어를 사용하여 검색을 수행할 수 있다. 뿐만 아니라 정확한 매칭을 요구하는 키워드 매칭 인덱스 기법보다 자연어를 이용함으로써 사용자의 편리성과 신속한 결과 얻을 수 있도록 성능을 향상 시켰다.

ABSTRACT

Currently users are tend to use multimedia data types in their applications. Main features of multimedia data types are large amount of data compared to conventional data types. In this reason, it's hard to load data into main memory and to search. That is the cause of occur disk input and output frequently, and decrease the system performance. In this paper, we describe to have fast and efficient access to multimedia data using index technique. Index method presented by paper consists of two parts : one is index file part for keywords and the other is posting file part for the list of file names. Of course, we use keyword. But user is not charge of memory for the keywords. Users just use natural language to insert, delete and search data what he or she wants. Internally, System makes keywords from natural language to get access to multimedia data. It provides convenience to users. Using this study to develop one's application for multimedia , one may have a chance for advanced performance of a system and getting a result speedily.

키워드

멀티미디어, 정보검색, 자연어처리, 인덱싱

*한세대학교 IT학부 조교수

**조선대학교 자연과학대학 컴퓨터통계학과 교수

접수일자 : 2004. 4. 28

Ⅰ. 서 론

멀티미디어 데이터 관리를 위하여 기존의 고정된 크기의 데이터 관리와는 달리 현재의 데이터베이스 관리 시스템 개발은 멀티미디어 시스템에 적합하도록 저장 구조의 물리적 측면에서의 투명성(transparency), 인덱싱을 통한 관련 데이터들의 접근, 정의된 접근 방법을 통한 데이터 일치성(consistency), 동시성 제어(concurrency control)을 통한 다중 사용자 접근, 회복 기법을 통한 신뢰성 등 다양한 서비스를 제공하기 위해서 멀티미디어 데이터들은 멀티미디어 데이터베이스 관리 시스템(MMDBMS)에 통합되어져야만 한다.

멀티미디어 데이터는 여러 페이지에 저장될 정도로 많은 기억 장치 공간을 요구하면서 많은 종류의 데이터 타입으로 인하여 사용자가 멀티미디어 데이터 요청시 빠른 접근을 허용해야 한다. 이를 위해서 이미지나 웨이브 데이터 등의 찾고자 하는 데이터 타입이 모두 주기억장치에 상주되어 있어야 하지만 제한된 크기의 주기억장치가 사용되므로 오히려 큰 데이터일 수 록 디스크와의 잦은 입출력으로 인하여 시간 낭비만 초래하게 된다. 그러므로 시간 낭비를 줄이고 빠른 접근이 이루어지도록 이들 멀티미디어 데이터들에 대해서 인덱스를 구성하고, 이 인덱스를 통한 접근이 필요하게 된다.

사용자가 인덱스를 구성하기 위해서 멀티미디어 데이터 접근에 필요한 키워드를 생성하게 되면 많은 수의 데이터들에 대해 모두 키워드를 정확하게 기억하고 있어야만 한다. 그래야 검색이 가능하게 될 것이다. 따라서 멀티미디어 시스템에서는 사용자의 정확한 키워드 매칭과 상관없이 원하는 정보를 검색할 수 있는 환경을 제공하여야 한다. 그러므로 멀티미디어 데이터의 내용을 기술한 문서를 기반으로 검색이 이루어질 수 있도록 하여야 하며, 또한 시스템이 자동적으로 키워드를 검출해 낼 수 있는 기능을 갖추도록 하여야만 한다.

그러므로 데이터베이스 내에 문서를 포함한 멀티미디어 데이터들에 대한 상호 관련성을 충분히 고려하여 저장되어지고 검색 관리되어지기 위해서 문서를 기반으로 하는 검색 시스템이 필요하며, 이에 본 연구에서는 사용자가 다루는 멀티미디어 데이터의 크기가 상당히 큰 크기를 요구하는 데이터인 경우에 보다 쉽고 빠르게 접근할 수 있도록 접근 방법에 관하여 살펴본다.

본 논문의 구성은 제 2장에서 현재 멀티미디어 검색 기술에 관한 관련 연구를 살펴보며, 제 3장에서는 본 논문에서 살펴보게 될 멀티미디어 데이터 검색 시스템의 기술에 관하여 살펴본다. 제 4장에서는 멀티미디어 영상 데이터에서 내용을 기반으로 검색이 이루어지는 시스템들과의 비교 분석을 살펴본다. 제 5장에서 결론으로서 현재 멀티미디어 검색 시스템의 기술상의 문제점 및 향후 연구 개발의 필요성이 요구되는 부분들을 살펴본다.

Ⅱ. 관련 연구

현재 멀티미디어 데이터 정보 관리 시스템과 관련된 다양한 방법들이 활발히 연구되고 있으며, 특히 영상 데이터베이스로부터 원하는 것을 찾는 영상 검색 방법은 새로운 분야로 각광받고 있다. 영상검색 기술은 영상을 분석하여 특징을 추출한 다음 이를 색인화하는 기술과 유사한 특징을 가지는 영상을 검색하는 기술로 텍스트 기반과 내용기반 영상검색 기법이 있다. 최근에는 색상, 질감, 모양 등의 특징들을 조합할 뿐 아니라 지식기반 시스템, 영상처리, DB관리 시스테, 정보 검색 시스템 등 다양한 분야에서 아이디어를 모색하고 있다[1,8].

대표적 영상검색 시스템으로 IBM에서 개발한 이미지 및 동영상 검색엔진 QBIC가 있으며[2], Virage사에서 개발한 Virage는 API를 제공하는 텍스트, 정지영상 및 동영상 검색엔진이다[3]. 또한 미국 Columbia 대학의 VisualSEEK는 인터넷에서 정보를 검색할 수 있으며, 초기화면에 저장하고 있는 데이터를 분류한 메뉴화면에서 주제별 검색을 수행 할 수 있다[4]. 그 외에 CHABOT 시스템[5]은 데이터의 내용을 기반으로한 데이터 검색과 데이터 저장을 위해서 또다른 저장 구조를 사용하는 시스템이다. 이 시스템은 디지탈화된 방대한 이미지 집합들에 대해서 저장 장소와 데이터에 대한 검색 기초로 연구가 시작되었다. 이 연구에서 다루게 되었던 이미지들은 사진과 같은 이미지들을 관리하는데 목적이 있었다. CEDAR(The Center of Exellence for Document Analysis and Recognition) 시스템인 CEDAR는 캡션된 이미지(captioned image)의 인덱싱과 데이터의 내용을 기반으로한 검색 시스템[7]이다. 이 시스템은 신문이나 잡지 등에서 부제목이 첨가된 사람들의 사진과 같은 이미지 검색을 필요로 하는 응용 분야를 위해 개발되었다.

특히, 부제목이 첨가되어 있는 이미지 데이터를 캡션된(captioned) 이미지라고 한다. 그러나 사람의 사진과 같은 그림이 삽입된 데이터베이스를 설계할 때 다음과 같은 문제점이 발생할 수 있다. 첫째로, 데이터베이스에 그림과 같은 이미지 데이터

를 삽입할 때 기존의 데이터가 차지하고 있던 크기보다 상당히 크므로 이때 처리해야 할양과 타입들에 관한 문제이다. 둘째는 질의 처리를 위한 효율적인 검색 스킴(scheme)에 관한 것이다. 이와 같은 부분은 기존의 데이터베이스에서 찾아보기 어려운 부분으로서 멀티미디어 시스템에서 다루어져야만 될 부분이다.

III. 멀티미디어 검색 시스템

1. 멀티미디어 데이터 구성

멀티미디어 데이터는 의미상에 있어서 많은 정보가 묵시적으로 정의되어 있다. 따라서 부가적인 데이터 설명이 없다면 실제로 데이터 내용에 의한 검색 방법은 성능 면에서 뛰어난 성과를 이룰 수 없다. 멀티미디어 데이터의 일반적인 구성은 2부분으로 이루어져 있다. 첫째로, 원시 데이터(raw data) 부분과 두 번째로는 데이터에 대한 내용을 기술한 기술 데이터 혹은 설명 데이터 (description data) 부분이다. 본 논문에서 제시하고 있는 데이터는 멀티미디어 데이터의 한 타입인 이미지 데이터의 예를 들었으며, 대부분의 멀티미디어 데이터 타입에 그대로 적용시킬 수가 있다.

일반적으로 원시 데이터 부분은 멀티미디어 데이터의 물리적인 특성을 지니는 실제 저장되는 데이터를 의미한다. 한 데이터 타입인 이미지 데이터 예에서 원시 데이터는 비트맵(bitmap) 형태로 표현된다. 이미지 멀티미디어 데이터 검색시 해당 데이터의 원시 데이터 부분이 실제로 검색된다. 원시 데이터 부분에 대한 검색이 이루어지려면 데이터의 크기가 크므로 한 번에 주기억장치에 기억되지 않을 경우가 발생하여 디스크 입출력으로 인한 시간적 손실이 생길 수 있다.

그러므로 본 논문에서는 멀티미디어 데이터 검색을 위하여 멀티미디어 데이터 구성에 부가적인 정보를 가지는 기술 데이터 부분을 추가한다. 기술 데이터 부분은 멀티미디어 데이터의 내용을 표현하는 데이터를 의미한다. 예를 들면, " two man standing by a tree colored rose " 이것은 이미지 데이터에 대한 기술 문장이라고 할 수 있다. 이때 기술 문장에 대한 내용은 사용자가 작성하므로 사용자가 결정한다.

2. 멀티미디어 데이터 검색 구조

멀티미디어 데이터들에 대한 검색 알고리즘은 대부분의 정보를 저장하고 있는 데이터베이스에서 사용자가 원하는 정보들을 가능한 빠르게 찾는 방법이다. 이들 검색 알고리즘들은 인덱스 활용에 따라 크게 두 가지 형태의 검색 알고리즘으로 구분지을 수 있다. 첫째로 멀티미디어 데이터 파일들에 대해서 순차적으로 스캐닝(scanning)을 하는 것이다. 이 방법은 저장되어 있는 모든 파일을 대상으로 작업이 이루어진다.

두 번째 방법으로는 멀티미디어 데이터에 대한 인덱스가 활용 가능하여 검색 속도를 향상시킬 수 있도록 인덱스된 텍스트를 사용하는 것이다. 이것은 인덱스 생성에 필요한 오버 헤드가 발생할 가능성과 인덱스 크기가 데이터베이스의 크기와 비례하여 커진다는 단점이 존재할지라도 검색 속도를 크게 향상시킬 수 있는 장점이 있다. 본 논문에서는 멀티미디어 데이터 검색을 효율적이며, 시간적으로도 상당히 단축할 수 있는 두 번째 방법으로 멀티미디어 데이터 검색 시스템을 설계 및 구축한다.

시스템 구축을 위해 인덱스로 사용되는 자료 구조로 우리가 흔히 접할 수 있는 B 트리를 이용하였다. 트리내의 한 노드는 디스크의 입출력을 고려하여 블록 단위의 노드로 전체 트리가 그림 1에 나타나 있다.

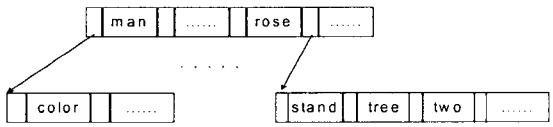

그림 1. 멀티미디어 데이터 검색을 위한 트리 구조

그림 1의 트리상에 나타난 노드내의 키워드들은 이전 절에서 언급하였던 데이터를 그대로 사용한 "two man standing by a tree colored rose"이다. 하나의 멀티미디어 데이터에 포함된 기술 문서에서 모든 단어가 키워드로서 작용하지 않고 그림 1에 나타난 키워드들처럼 단지 몇 개의 단어만 키워드로서 나타나 있다. 위의 예제에서 파일 내의 기술 문장 가운데 "standing"에서 "ing"와 "by", "a", "colored" 의 "ed"등은 단어가 무의미하거나 변형된 단어인 관사, 변형된 낱말에 해당되므로 키워드 매칭 비교 횟수를 줄이기 위해서 제외시킨다. 이에 대한 설명은 다음의 연산 동작들에서 좀더 자세히 살펴본다.

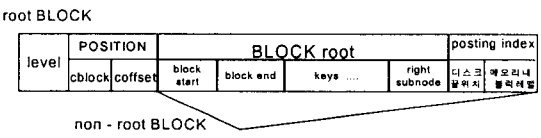

그림 2. 트리상의 노드 구조

그림 2은 그림 1에 나타나 있는 한 노드의 구조를 도시하고 있다.

그림 2의 한 노드 구조에서 앞쪽에 위치한 레벨은 인덱스 트리내의 레벨을 의미하며, 포지션의 "cblock"과 "coffset"은 캐쉬 내에 존재하는 블록에 관한 정보이다. 다음의 "BLOCK"은 실제 데이터들이 삽입될 영역이다. "block start" 와 "block end"는 블록 내에 존재하는 키워드들이 기억되기 시작한 메모리 위치와 끝 위치를 나타낸다. "keys"는 여러 개의 키워드가 위치할 수 있는 부분을 명시하고 있으며, "right subnode"는 루트 노드를 기준으로 우측 서브 노드의 위치를 명시한다. 다음의 포스팅 인덱스 부분은 포스팅 파일에 대한 정보를 기록하고 있다. 포스팅 인덱스에서 디스크 끝위치는 새로운 키워드 입력시 이에 대한 블록 영역을 할당받게 될 위치를 나타내는 것이다. 메모리 내의 블록 레벨은 메모리 상에 존재하는 포스팅 파일의 레벨을 명시한다. 일반적인 B 트리의 특성에서 리프(leaf) 노드와 리프노드가 아닌 비리프(non-leaf) 노드들의 구조가 서로 다르다. 그러나 본 논문에서 리프노드에 해당되는 부분이 포스팅 파일이며 비리프 노드가 트리를 구성하고 있는 인덱스 파일이다. 루트 노드와 비루트 노드는 그림 2에서 알 수 있듯이 구조가 서로 다르다.

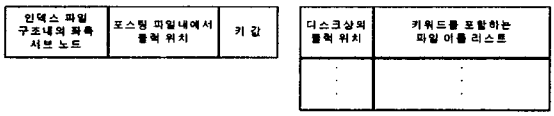

그림 3. Keys를 구성하는 키워드구성(좌)과 포스팅 파일 구조(우)

그림 3에서는 그림 2의 "keys"를 구성하는 키워드들 중 하나의 키워드 구조를 좌측에 나타내고 있다. keys의 구성에서 첫 번째 부분은 좌측 서브 노드를 가리키며, 가운데 부분은 해당 키워드를 파일내의 기술 데이터 부분에 가지고 있으며, 파일 이름들을 저장하고 있는 포스팅 파일 내의 위치를 나타낸다. 마지막 부분에 저장되어 있는 키값이 실제로 인덱스로 동작하기 위한 키워드이다. 그림 3에서 오른쪽 그림은 키워드를 가지는 파일 이름 리스트와 포스팅 파일 내에서의 메모리 위치로 구성되어 있다.

이와 같이 그림 1, 2, 3을 통하여 인덱스를 구성하는 구조들을 살펴보았다. 일반적인 인덱스 트리에서는 키워드만을 기록하며 키워드가 어느 파일들에 속해 있는지를 알 수 없다. 그러므로 위의 그림들에서 알 수 있듯이 파일 이름들의 리스트를 유지하기 위해서 포스팅 파일을 따로 관리하도록 한다. 왜냐하면 트리상에서 파일 이름을 관리할 경우에 한 노드내에 존재할 수 있는 키워드의 개수가 상대적으로 적어지며, 이로 인하여 트리 레벨이 높아지고, 키워드 탐색시 높은 레벨 때문에 노드 탐색 회수가 많아진다. 뿐만 아니라 잦은 디스크 입출력 발생의 원인이 된다. 이와 같은 이유 때문에 키워드로서 존재하는 부분과 파일들의 리스트를 따로 관리하여 디스크의 입출력을 최대한 줄이도록 하였다.

3. 멀티미디어 데이터 정보에 대한 기술 데이터의 키워드 생성 과정

키워드를 추출하기 위해 기술 데이터에서 사용자에게 자연어는 사용성이 용이하므로 자연어를 이용한 데이터의 삽입, 삭제 및 갱신 연산을 수행하는 것이 사용자의 편리성을 향상시킬 수 있게 된다. 그러나 이들 자연어들은 직접 키워드로서 사용할 수 없으므로 자연어를 키워드에 맞도록 변환 과정이 필요하다. 변환을 거쳐서 생성된 키워드는 멀티미디어 데이터베이스 내에 저장되게 된다. 이를 위한 키워드 생성 과정이 그림 4에 도시되어 있다. 그림 4에 대한 내용은 멀티미디어 데이터 검색 연산 수행을 통하여 좀더 자세히 살펴본다.

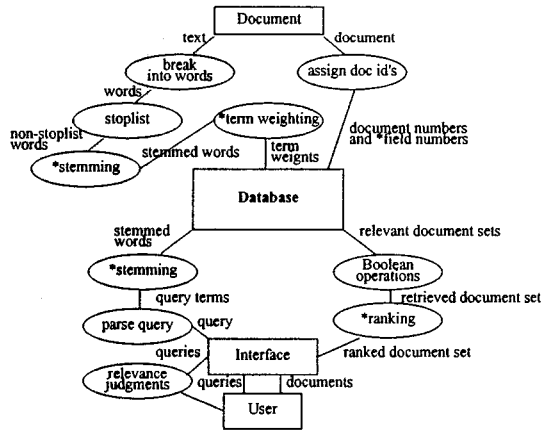

그림 4. 멀티미디어 정보 검색 시스템 흐름도

4. 멀티미디어 데이터 정보에 대한 연산 수행

본 절에서는 가능한 자연어를 사용하여 멀티미디어 데이터 정보에 대한 내용을 기술하고 이들 데이터에 대해, 신속한 삽입, 삭제, 갱신 등의 연산이 수행되는 과정을 그림 4를 통하여 살펴본다.

4.1 멀티미디어 데이터 삽입 연산

삽입 과정은 멀티미디어 데이터를 저장할 때 자연어에서 발생할 수 있는 모호성을 배제하고 키워

드와 매칭을 간략화 하기 위해서 파서(parser)에 의해 낱말 단위로 변환된다. 이를 위한 동작은 하나의 파일 내에 멀티미디어 데이터 정보를 기술한 문서가 어휘 분석기를 통하여 하나 하나의 낱말 단위로 분리되도록 하는 것이다. 이때 문서를 구성하는 문장은 사용자의 편의를 도모하기 위하여 자연어와 같이 일상적인 낱말들로 구성된 문장을 입력하게 된다. 이들 낱말 중에는 키워드로써 사용 가능한 낱말이 존재할 수 있고, 혹은 관사와 같이 중요한 의미가 부여되지 않아서 키워드로써 사용하지 않아도 되는 낱말들이 있다. 이들을 구분하여 키워드로서 사용할 수 있는 낱말들만을 선택하는 과정이 필요하며, 이러한 과정이 "stoplist" 부분이다. stoplist 과정을 거친 낱말들은 아직 키워드로서 사용할 수 없는 상태이며 낱말이 가지는 본래의 형태인 루트는 아니다.

이들 각각의 낱말들을 낱말의 원형으로 바꾸어 인덱스에 사용하기 위해서 낱말의 형태를 원형으로 바꾸는 "stemming" 과정을 거치도록 한다. 이 과정을 거친 낱말들은 키워드로서 자격을 갖춘 낱말이 되므로 인덱스 파일에 삽입이 이루어질 수 있다. 만약, 좀더 세분화된 과정을 거친다면 "term weighting"이라는 부분에서 stemming 된 각 낱말들에 대해 중요도에 따라 가중치를 두어 데이터베이스에 삽입하도록 한다. 이때 인덱스 파일 내에 삽입되는 키워드는 해당 키워드가 어느 파일 내에 속하는지를 알고 있어야한다. 이를 위해서 키워드가 속한 파일 이름이 포스팅 파일에 저장되며 키워드는 B 트리 구조를 가지는 인덱스 파일에 저장하게 된다.

4.2 데이터 검색 연산

검색 연산은 삽입 연산을 이용하여 저장된 아이템들에 대해 자연어로 질의를 입력하고, 시스템이 질의 내의 키워드를 이용해서 인덱스를 통한 데이터 검색을 수행한다. 이렇게 수행된 검색 결과로서 해당 파일 이름을 반환하는 연산을 검색 연산이라 한다. 반환된 결과를 바탕으로 정확하게 원하는 데이터인지를 확인하는 것은 사용자가 판단한다.

사용자가 원하는 것은 멀티미디어 데이터 파일이다. 그러므로 크기가 큰 멀티미디어 데이터를 보다 빠르게 찾기 위해서 인덱스를 검색한다. 또한 인덱스로 사용되는 B 트리 구조 내에서 키워드 매칭이 이루어져야 한다. 키워드는 키워드가 속한 파일들의 리스트를 유지하기 위해서 포스팅 파일을 가리키고 있어야 한다. 본 논문의 검색 과정이 이루어지기 위해서는 먼저 전체 트리에 존재하는 각 노드들에 대한 탐색이 선행되어야 한다. 다음으로 노드내에 포함되어 있는 키워드들에 대한 비교 매칭이 이루어진다. 매칭이 이루어진 키워드에 대해 포스팅 파일 내의 파일 리스트를 반환한다. 이 반환 결과에는 여러 개의 파일들이 존재할 수 있다. 그러므로 사용자가 원하는 파일을 보다 정확하게 탐색하기 위해서 각 키워드로부터 반환 받은 파일들에 대한 추가적인 연산을 수행한다. 이 부분은 사용자가 작성한 질의 문의 조건을 모두 만족할 수 있도록 파일을 AND/OR 시켜서 반환해 준다. 이와 같은 검색 연산은 보다 신속하고, 정확성과 편리성을 향상시키고 사용자에게 편리한 정보 검색 환경을 제공해 주는 역할을 할 수 있게 된다.

4.3 데이터 삭제 연산

삭제 연산은 일반적인 데이터베이스 내에 인덱스 구조로 사용하는 B 트리에서의 삭제 연산과 동일하다. 단지 삭제 연산을 위하여 입력받은 데이터는 삭제하려는 파일 이름이 된다. 이 파일 이름을 기반으로 먼저 파일 내에 존재하는 멀티미디어 데이터의 기술 데이터 부분을 가져와서 이 부분에 대해 키워드를 추출하고 추출한 키워드들에 대해 포스팅 파일 내에서 해당파일 이름을 삭제한다. 즉, 입력 및 검색 연산을 수행하던 과정처럼 "stoplist"와 "stemming" 과정을 통해서 키워드를 추출하는 연산이 동일하게 요구된다. 이 과정을 거친 어휘들이 키워드로서 가치가 있으므로 이들에 대해 데이터베이스 내에서 삭제 연산이 이루어지도록 한다.

한 키워드에 대해서 포스팅 파일 내에 있는 파일 이름 삭제시 고려해야 할 사항이 있다. 인덱스 파일에 있는 키워드가 파일 이름이 저장된 포스팅 파일 위치를 가리킨다. 이 포스팅 파일의 한 블록 내에 하나의 파일만 존재할 경우와 여러 개의 파일이 존재할 경우가 발생한다. 첫 번째 경우에는 포스팅 파일의 한 블록에 하나의 파일만 존재하므로 파일 이름의 삭제와 인덱스 파일에서 키워드 삭제가 병행해서 수행되어야 한다. 이에 반하여, 두 번째의 경우는 포스팅 파일 내에 여러 개의 파일 이름이 존재하므로 해당 파일 이름만 삭제 하므로서 삭제 연산이 이루어지게 된다.

IV. 비교 분석

멀티미디어 데이터베이스에서의 데이터 검색은

상당히 어려운 문제이다. 특히 기존의 데이터와는 비교가 되지 않을 정도로 크기가 큰 멀티미디어 데이터를 관리하고, 또한 정의하기 힘든 멀티미디어 데이터의 내용을 기반으로 관련 데이터를 신속하고 효율적으로 검색하여야 하기 때문이다. 본 절에서는 논문에서 제시한 멀티미디어 정보 검색 기법을 기존의 연구에서 살펴본 멀티미디어 정보 검색 시스템들과 비교하여 차이점과 본 연구의 장점 등을 살펴본다.

본 논문에서의 멀티미디어 데이터 구성은 영상 이미지나 오디오, 비디오 등의 멀티미디어 데이터 타입과 이들 멀티미디어 데이터들을 설명할 수 있는 기술 데이터 부분으로 구성되어 있다. 그러나 기존의 연구들에서는 특정 응용 프로그램에 맞게 멀티미디어 데이터를 구성하였고 이들 멀티미디어 데이터의 내용 기술 데이터 활용이 미흡하였다. 그러나 본 논문에서의 멀티미디어 데이터는 임의의 멀티미디어 데이터에 국한되지 않고 모든 멀티미디어 데이터 타입에 그대로 적용시킬 수 있다. 또한 일반적인 정보 검색 분야에서는 내용 설명을 키워드로 표현한다. 그러나 이들 키워드들로 표현할 때 단점은 정확성이 떨어지며 모호성이 증가된다. 특히 사용자가 키워드를 정확하게 일치시켜야만 데이터 탐색이 이루어지지만 본 논문에서 제시한 자연어를 이용하면 키워드를 암기해야 하는 불편을 제거할 수 있다.

이미지 등을 많이 다루는 응용 프로그램들에게서 내용을 기반으로 키워드를 미리 선언하여 사용자가 이 내용을 기반으로 한 키워드를 활용하는 멀티미디어 검색 시스템의 개발이 이루어지고 있다. 그러나 사용자가 멀티미디어 데이터 탐색시 내용 기반 키워드를 선언하는 것은 본 논문에서 자연어를 이용하여 멀티미디어 데이터 내용을 기술할 때 사용되는 낱말과 큰 차이가 없다. 뿐만 아니라, 영상 멀티미디어 검색 시스템에서 이미지를 검색하는 동안 시스템의 오버 헤드와 시간 낭비를 막을 수 있다.

V. 결 론

오늘날 사용자가 요구하는 데이터들은 대부분 멀티미디어 데이터들로서 영상, 그래픽, 소리, 애니메이션, 비디오, 오디오등이 해당된다. 이들 데이터를 필요로 하는 사용자 응용 프로그램들은 기존의 정형 데이터와 같은 고정된 크기의 데이터 타입만으로 요구에 적절하게 대처할 수 없게 되었다. 그러므로 이들 요구에 대응하기 위해서 정형화된 기존 데이터 형태 뿐만 아니라 멀티미디어 데이터 형태도 함께 포함된 데이터 형태가 필요하게 되었다.

본 논문에서는 다양한 형태의 멀티미디어 데이터들 중 데이터에 대한 내용을 기술한 문서가 파일에 함께 포함되어 있을때 보다 효율적이며 신속하게 검색할 수 있는 방법을 연구하였다. 이를 위해서 멀티미디어 데이터베이스 내에 사용될 인덱스 구조로 B 트리를 사용하였으며, 사용자의 편리성을 향상시키기 위하여 사용자가 인터페이스를 통하여 데이터 입력 및 검색 연산을 위한 질의문 요청시 일반적으로 사용하는 자연어 같은 형태로 파일 내용을 기술, 저장 및 검색할 수 있도록 하였다. 그러므로 본 연구를 활용하면 도면과 같은 이미지와 이미지 내용을 포함한 문서가 하나의 파일로 취급되는 응용 프로그램인 경우에 검색의 효율성을 충분히 기대할 수 있을 것이다.

향후 연구로서 데이터의 크기가 큰 멀티미디어 데이터 관리 및 관련 연구 분야에서 데이터 입출력시 한 페이지보다 큰 기억 공간을 요구한 데이터 처리와 사용자의 동시 접근에 따른 동시성 제어 문제에 관하여 연구가 필요할 것이다. 뿐만 아니라, 데이터 파손에 따른 복구시 로깅 기법으로 해결하기에 많은 오버헤드가 초래되기 때문에 이들에 관한 연구가 필요하다.

참고문헌

[1] Kozaburo Hachimura, "Retrival of Paintings Using principal Color Information", IEEE Proceedings of ICPR, 1996.

[2] H.V. Jagadish, "A Retrieval Technique for Similar Shapes", Proceedings of the ACM SIGMOD International Conference on the Management of Data, pp. 208-217, May 1991

[3] Myron Flicker, Harpreet Sawhney, Wayne Niblack, Jon Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video

content: the qbic system. IEEE Computer, 28(9):23-32, September 1995.

[4] R.K. Srihari and D.T. Burhans, "Visual Semantics : Extracting Visual Information from Text Accompanying Pictures", Proc. AAAI94, American Association for Artificial Intelligence, Menlo Park, Calif., 1995, pp. 793-798.

[5] E. Ogle, Michael Stonebraker, "Chabot : Retrieval from a Relational Database of Image", IEEE Computer, 28(9):40-48, September 1995.

[6] Chung-Sheng Le, Rakesh Mohan and John R. Smith, "Multimedia Content Description In the InfoPyramid," IEEE ICASSP, Vol.6, pp3789-3792, 1998

[7] Rohini K. Srihari, "Automatic Indexing and Content-Based Retrieval of Captioned Images", IEEE Computer, 28(9):49-56, September 1995.

[8] Ajay divakaran, Hirofumi Nishikawa, Kohtaro Asai, "A Description Scheme For Video Based On Feature Extraction In the Compressed Domain", IEEE, pp278-279, 2000

저자소개

이현창(Hyun-Chang Lee)

1993년 원광대학교 컴퓨터공학과(공학사)
1996년 홍익대학교 전자계산학과(이학석사)
2001년 홍익대학교 전자계산학과(이학박사)
2001.3～2003.8 경인여자대학 IT학부 조교수
2003.9～현재 한세대학교 IT학부 조교수
※관심분야 : 멀티미디어 시스템, 주기억 데이터베이스 시스템, 웨어하우스, XML

배상현(Sang-Hyun Bae)

1982년 조선대학교 전기공학과(공학사)
1988년 일본동경도립대학 전기정보공학과(공학박사)
1995년 일본과학기술원 전자정보통신공학부 초빙교수
1998년 일본동경도 멀티미디어 연구소 객원교수
2002년 캐나다 Univ of Alberta 전자계산학과 객원교수
1988년～현재 조선대학교 자연과학대학 컴퓨터 통계학과 교수
※관심분야 : 대규모 지식베이스, 인공신경망, 이동에이전트