
영상 캡션 정보를 이용한 멀티미디어 데이터 검색 시스템의 설계 및 구현

이현창* · 배상현**

Design and Implementation of Multimedia Data Retrieval System using Image Caption Information

Hyun-chang Lee* · Sang-hyun Bae**

요 약

오디오 비디오 데이터의 활용이 증가함에 따라 멀티미디어 데이터의 내용에 대해 표현하려는 연구와 함께 멀티미디어 데이터의 내용이나 메타데이터를 저장하고, 검색하고, 조작하는 연구의 필요성이 증가하였다. 멀티미디어 데이터 표현은 사용자가 원하는 내용만을 쉽게 검색하고, 접근할 수 있도록 표현되고 저장되어야 한다. 본 논문은 멀티미디어 데이터가 포함된 문서를 검색하기 위해서 멀티미디어 데이터의 캡션(Caption) 정보를 이용하거나 문서 내용을 기반으로 멀티미디어 데이터를 검색할 수 있는 시스템을 설계하고 구현하였다. 멀티미디어 데이터의 캡션 정보와 문서의 텍스트는 모두 키워드를 추출하기 위해 필터링(Filtering) 단계를 거치고, B+트리를 이용한 역 파일 구조를 사용하여 빠르고 대용량의 문서 검색을 할 수 있도록 하였다.

ABSTRACT

According to the increase of audio and video data utilization, the presentation of multimedia data contents and the work of retrieving, storing and manipulating a multimedia data have been the focus of recent work. The display for multimedia data should retrieve and access the contents easily that users want to present. This study is about the design and implementation of a system to retrieve multimedia data based on the contents of documentation or the caption information of a multimedia data for retrieving documentation including multimedia data. It intends to develop an filtering step to retrieve all of keyword within the caption information of multimedia data and text of a documentation. Also, the system is designed to retrieve a large amount of data quickly using an inverted file structure available for B+ tree.

키워드

멀티미디어, 캡션정보, 영상, 정보검색

*한세대학교 IT학부 컴퓨터공학과 조교수

**조선대학교 자연과학대학 컴퓨터통계학과 교수

1. 서 론

우리가 일상 생활에서 많이 사용하는 문서들은 그 목적이 우리에게 정보를 전달해 주는 도구로서의 역할을 한다. 이런 문서들의 기본 구성은 주로 인간이 표현할 수 있는 모든 것들을 기록할 수 있는 텍스트와 이해를 돕기 위한 사진, 그림과 같은 영상들을 포함하는 멀티미디어 데이터들로 구성될 수 있다. 실제로 우리 주변에는 너무나 많은 문서들이 존재하기 때문에 우리가 원하는 문서를 찾기란 쉽지 않다. 모든 문서들을 조사하는 것은 비효율적인 일이기 때문에 이를 해결하기 위한 많은 방법들이 연구되어왔다.

컴퓨터의 출현으로 인해 이를 이용하여 대용량의 문서들은 효과적으로 검색할 수 있는 정보 검색 시스템들에 관한 연구가 많이 이루어지고 있다 [1,2,3,4]. 이로 인하여 사람이 일일이 수작업으로 문서들을 나누어 검색하는 방법에서 컴퓨터를 이용한 색인으로의 검색이 이루어졌으며 사용자가 원하는 정보에 대하여 신속하고 정확하게 전달해 줄 수 있게 되었다.

최근에는 컴퓨터의 발달로 대용량의 데이터를 저장할 수 있게 되었고, 텍스트 뿐만 아니라 이미지, 음성, 비디오 등 용량이 매우 크고 비정형화된 데이터들을 저장할 수 있게 되었다. 이로써 이전의 텍스트와 같은 정형화된 데이터와 비정형화된 멀티미디어 데이터를 함께 가지고 있는 문서에 대한 효율적인 검색이 필요하게 되었다. 예를 들면 사진이 포함된 신문 잡지의 기사 검색이나 특허 정보, 도면의 검색을 들 수 있다.

특히 멀티미디어 데이터들 중 영상 관련 데이터를 검색하기 위한 많은 방법들이 제안되었다. 이미지 데이터의 내용을 기반으로 한 검색 방법은 패턴 인식 분야에서 오랫동안 연구가 되어왔던 문제이다. 내용에 기반한 이미지 데이터의 검색은 매우 다양한 실세계의 이미지를 정확하게 인식하기 어렵고, 그 구현이 복잡하다는 단점을 가지고 있다 [6]. 그러므로 이미지 멀티미디어 데이터를 설명하기 위한 설명 정보의 요구가 절실하다.

본 논문에서는 기존의 문서와 이미지에 대한 검색이 다른 기법들에 의하여 따로 설계되어왔기 때문에 기존의 텍스트 내용 기반의 검색 방법을 이용하여 이를 통합하였다. 내용 기반의 검색을 위해 문서는 텍스트 데이터를 사용하고, 이미지는 캡션 정보를 이용하여 동시에 문서의 접근이 가능하도록 설계하고 구현하였다.

본 논문은 2절에서 full-text 검색 방법과 멀티미디어 데이터를 위한 이미지 검색 방법, 그리고 파

일 구조에 대한 연구를 살펴보고, 3절에서는 검색 시스템의 설계와 구현을 위한 필터링 방법과 색인 파일 그리고 포스팅 파일에 대해서 설명한다. 또한 문서와 이미지의 내용을 이용하여 동시에 문서를 검색하는 방법에 대해서 기술한다. 4절에서는 결론과 향후 연구 과제에 대해 기술하였다.

II. 관련 연구

1. Full-Text 검색 방법

데이터를 검색하는데 있어서 가장 중요한 것은 검색하고자 하는 사용자의 질의와 저장된 데이터를 일치시키는 일이다. 초기의 컴퓨터를 이용한 문서의 검색 방법은 full-text 문서를 주로 다루었으며 각 문서의 입력시 적당한 키워드를 주거나 텍스트 묘사(text description)를 이용해서 검색하였다. 이런 심볼 묘사(symbolic description) 방법은 데이터를 묘사하기 위한 적절한 키워드를 입력하는 것이 무엇보다도 중요하며 질의에 대한 키워드와 저장된 데이터의 키워드가 일치하지 않을 수 있다는 단점을 가지고 있다. 이를 보완하는 방법으로 다중 키워드(multiple keyword)를 이용한 검색 기법을 사용하였다[5]. 다중 키워드 기법은 단어 사전을 이용하여 사용자의 질의에 대한 키워드를 단어 사전에서 유사한 뜻을 지닌 단어를 찾고 이런 다중 키워드에 대한 모든 데이터의 검색을 요구하는 기법이다.

2. 이미지 데이터 검색 방법

이미지 데이터를 검색하는 방법은 이미지 데이터와 관련된 키워드를 사용하는 방법으로써 사용자가 이미지 데이터를 입력할 때 적절한 키워드를 같이 입력해야 한다. 이것은 키워드의 선택에 있어서 사용자의 질의와 일치시키기가 어렵다. 이런 키워드에 관계없이 이미지 데이터의 내용을 기반으로 한 검색 방법으로 이미지의 특징을 시스템이 자동으로 추출하여 인덱싱을 만드는 방법으로 사용자가 원하는 이미지 데이터를 내용을 기반으로 검색할 수 있다. 이미지의 특징 추출 과정은 에지 검출 단계, 잡음 제거 단계, 객체 영역 추출 단계, 크기 정규화 단계, 영역 분할 단계, 특징 추출 단계 등으로 구성된다[3, 4].

내용 기반 데이터 검색을 위한 시스템으로는 사진과 같은 이미지를 관리하기 위하여 이미지의 고유번호를 가지고 검색을 할 수 있는 CHABOT 시스템이 있는데 이것은 나중에 온라인 브라우저를

지원하고 내용으로 이미지를 검색할 수 있도록 확장되었다[7]. CEDAR(The Center of Excellence for Document Analysis and Recognition) 시스템은 부제목이 첨가된 사람의 사진들에 대해서 인덱싱과 데이터 검색을 필요로 하는 응용 분야를 위해 개발되었다[9].

3. 파일 구조

텍스트에 대한 빠른 검색을 위한 여러 파일의 기법들이 제안되었으며, 본 내용에서는 요약 파일과 역 파일에 대해서 간략히 살펴본다.

3.1 요약 파일(Signature File)

Faloutsos는 텍스트 검색 기법의 연구로 텍스트 데이터에 적용할 수 있는 접근 기법으로 요약 파일 기법을 제안하였다[11]. 요약 파일 기법은 해쉬 함수에 의하여 검색 단어들을 비트열로 만들어 질의에 만족할 가능성이 있는 문서들만을 선택하여 접근함으로써 검색 시간을 줄이는 기법이다. 각 검색 단어의 비트열들을 중첩하여 만족하는 문서들만을 먼저 접근한다. 이 기법은 역 파일 방법에 비해서 부가 저장 공간은 적지만 검색 속도는 느리다는 단점을 가지고 있다. 이와 같은 단점을 보완하기 위한 방법으로 요약 파일 기법에 대한 검색 시간을 줄이기 위해서 특별 부호화 기법, 비트 슬라이스 기법, 다단계 인덱스 기법 등이 연구되었다.

표 1. 요약 생성 과정 예

Table 1. Example for a process of creating signature

데이터	해싱값
information	0110 0000 0000
retrieval	0000 0000 0011
system	0000 1001 0000
블록 요약	0110 1001 0011

3.2 역 파일(Inverted File)

역 파일 기법은 각 키워드와 관련된 문서들에 대한 링크를 포함하는 배열구조로서 B-트리, Prefix B-트리, 트라이 구조를 이용해서 구현할 수 있다. 역 파일 기법은 요약 파일에 비해 많은 부가 저장 공간이 필요하지만 검색 속도는 빠르다는 장점을 가지고 있다. 그리고 대용량의 문서들을 검색할 때 유용하며 이해하기 쉬워 구현하기가 쉬운 장점이 있다.

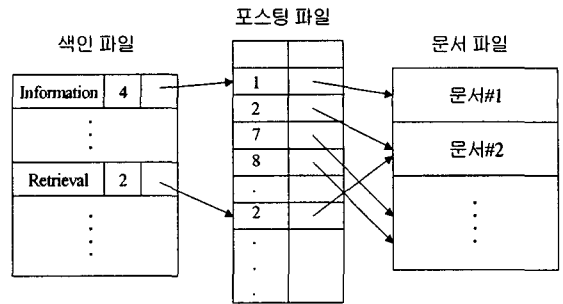


그림 1. 역 파일 구조
Fig 1. Inverted file Structure

III. 문서와 이미지 검색 시스템 설계 및 구현

1. 전체 시스템의 구성

본 연구에서 살펴볼 전체 시스템의 구성은 그림 2와 같다. 이미지가 포함된 문서로부터 이미지의 캡션 정보와 문서의 내용으로부터 텍스트를 따로 Scanning 처리 단계를 수행한다. 스캐닝 단계를 마치고 나서 필터링 단계에서 키워드를 생성하여 이미지와 문서에 대한 각각의 색인 파일과 포스팅 파일을 생성 및 저장한다. 사용자의 질의는 색인 파일과 포스팅 파일을 이용해서 문서와 멀티미디어 데이터인 이미지의 내용을 동시에 검색할 수 있게 된다.

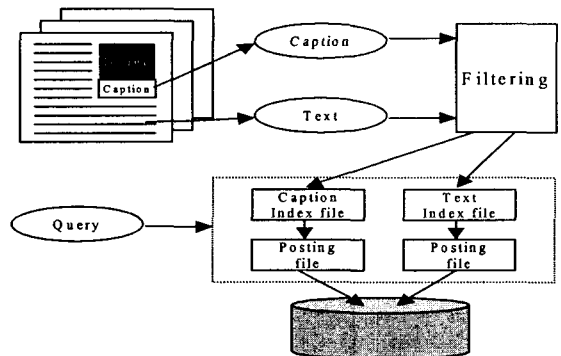


그림 2. 전체 시스템 구성도
Fig 2. System configuration

2. 필터링(Filtering) 단계

필터링 단계에서는 인덱스 파일에 저장될 키워드를 텍스트 문서로부터 자동으로 생성시키는 단

계이다. 본 단계는 인덱싱을 위한 전처리 단계라고 할 수 있으며 어휘 분석 단계, 불용어 제거 단계, 스테밍 단계, 단어 가중치 부여 단계, 색인 파일과 포스팅 파일에 저장하는 단계로 구성된다. 다음은 각 단계별 처리 내용에 관하여 간략히 살펴본다.

2.1 어휘 분석(Lexical analysis) 단계

어휘 분석 단계에서는 문서의 텍스트 내용을 각각 단어(term, word)별로 색인하기 위해 색인할 단어를 추출을 수행하는 단계이다. 구현을 하기 위해서는 일반적인 방법인 빈칸(white space)을 기준으로 단어를 추출하게 되고, 숫자와 빈칸, 탭 그리고 특수 문자들은 무시하여 수행하게 된다.

2.2 불용어(Stoplist) 제거 단계

불용어 제거 단계에서는 색인을 위한 단어들 중 키워드가 될 수 없는 단어들을 불용어라고 하며, 이를 제거하는 단계이다. 불용어는 키워드로서의 역할을 할 수 없기 때문에 본 시스템의 검색 효율을 높이기 위해 미리 제거되어야 한다. 불용어를 구현하는 데는 어휘 분석 단계의 결과 값을 다른 함수에서 처리하는 방법과 어휘 분석기내에 함수의 일부분으로 제거 함수 기능을 가지는 두 가지 방법이 있다. 본 연구에서 구현된 방법은 효율이 좋은 두 번째 방법으로 하였으며 불용어의 선택은 Francis and Kucera(1992)에서 정의된 425개의 단어를 기본으로 하였다.

2.3 스테밍(Stemming) 단계

본 단계에서는 어휘 분석 단계에서 반환된 단어들을 단어의 원형으로 만들어 주는 단계이다. 스테밍 단계는 단어의 크기를 줄여서 색인 파일의 크기를 줄여 주어 저장 공간의 낭비와 검색 효율을 향상시키게 도와주는 단계이다. 또한 스테밍 단계에서는 단어의 변형에도 검색시에 사용자 질의의 키워드와 색인 키워드의 일치 가능성을 해준다. 예를 들면, stemming, stem, stemmer 등의 단어를 stem으로 일치시켜주는 역할을 수행한다.

스테밍 단계 구현에서의 스테밍 함수는 5가지 규칙 리스트(rule list)를 가지고 있다. 그 중 해당하는 하나 이상의 규칙 리스트를 선택적으로 적용하는 것으로 이루어졌다. 이 규칙 리스트는 각각 복수형, 과거형, 명사형 등에 대한 단어의 원형을 만들어 주는데 필요한 정보를 가지고 있다. 예를 들면, 복수형 sses를 ss로 교체하거나 과거형 ed를 공백으로 하고, 명사형 ousness를 ous로 바꾸어 주는 임무를 수행하게 된다.

2.4 단어 가중치(Term weight) 부가 단계

단어 가중치 부가 단계에서의 역할은 단어에 가중치를 부가함으로써 사용자에게 질의의 결과를 반환할 때 중요한 단어들부터 정렬하여 보여줄 수 있도록 한다. 단어 가중치 부가는 부울린(boolean) 검색 모델의 문서 표현 방법을 향상시킨 벡터 공간 검색 모델에서 사용하는 방법이다. 단어 가중치는 $tf * idf$ 형태의 함수를 사용해서 계산되며, 이를 수행하기 위해서 단어 빈도수를 구하는 기능과 내포된 단어의 총 문서 수를 구하는 기능을 구축하였다. 예를 들면, tf (term frequency weight)는 문서 안에 그 단어의 빈도수를 구하는 함수이고, idf (inverse document frequency weight)는 그 단어를 포함하고 있는 총 문서의 수를 구하는 역함수이다.

3. 색인 파일(Index file) 구조

본 연구에서는 색인 파일 구조로 B+트리 구조를 사용하였다. B+트리는 다른 파일 구조에 비하여 약간의 메모리 공간을 더 요구하지만 갱신 연산에 대한 처리를 다른 색인 방법 보다 쉽게 수행할 수 있으며, 탐색 시간이 빠르다는 장점을 가지고 있기 때문에 B+트리를 사용하였다.

4. 포스팅 파일(Posting file)

포스팅 파일은 색인 파일의 단말 노드인 데이터 블록을 의미하며, 색인 파일이 가지고 있는 키워드와 연관된 문서에 대한 객체 식별자(Object identifier)를 가지고 있다. 객체 식별자는 문서에 대한 유일성을 나타낼 수 있어야 하므로 문서 파일에 대한 절대 경로와 파일의 이름으로 구성되어 있다. 한 개의 키워드에 해당하는 포스팅 파일에는 여러 개의 객체 식별자로 구성된 블록들을 가질 수 있다. 포스팅 파일의 블록 구조는 다음과 같다.

4.1 블록 구조

표 2. 포스팅 파일의 블록 구조
Table 2. Block structure of posting file

number	Oid1	Oidn	next
--------	------	-------	------	------

표 2에서 number 필드는 삽입된 Oid의 개수로 블록의 초기화나 Oid에 대한 검색 연산을 수행할 때 사용되며 현재 블록의 상태 즉, 블록의 오버플로어(overflow)나 언더플로어(underflow) 상태를 알 수 있기 때문에 블록의 관리에서도 사용된다. Oid 필드는 절대 경로를 갖는 파일이름의 객체 식별자

들의 리스트이다. Oid의 개수 n을 정할 때에는 디스크의 기억 장소와 검색 시간을 줄이기 위해 적절한 개수를 정해야 한다. 개수가 너무 크면 한 개의 키워드에 대한 블록의 크기가 커지므로 기억 장소의 낭비를 초래할 수 있고, 개수가 너무 작으면 빈번한 블록의 할당이 일어나게 되므로 검색 시간이 느려질 수 있기 때문이다. next 필드는 한 블록의 Oid가 오버플로어가 발생될 경우 새로 할당된 블록을 가리키는 디스크상의 오프셋을 나타낸다.

4.2 블록 관리

색인 파일은 한 개의 키워드에 대하여 디스크 상에서 한 포스팅 파일 블록의 오프셋을 가지고 있다. 그러므로 데이터 블록은 적어도 한 개의 Oid를 가지게 되고, 블록이 Oid를 하나도 갖지 못한다면 그 블록이 해당하는 오프셋은 다른 키워드의 블록의 재활당을 위해 자유 블록 리스트(free block list)에 등록된다.

새로운 블록이 필요할 경우 먼저 자유 블록 리스트에 등록된 블록을 먼저 할당 해주고, 만약 자유 블록 리스트에 데이터 블록이 존재하지 않을 경우 포스팅 파일의 맨 끝의 오프셋을 할당받게 된다. 이렇게 함으로써 블록의 빈번한 삽입과 삭제시에도 기억 장소의 낭비를 줄일 수 있다.

4.3 메모리내에서의 포스팅 파일의 블록 구조

디스크상에 저장된 포스팅 파일의 블록을 사용하기 위해서는 멀티미디어 데이터 검색을 위한 색인 파일들은 메모리에 적재되어야 한다. 그러므로 디스크 상에서 블록들의 연결이 정수형의 오프셋을 사용하지만 메모리 상에서는 블록의 연결이 메모리 포인터로 변환되어야 하기 때문에 블록의 구조가 달라질 수 있다. 메모리내의 블록 구조는 다음 표3과 같다.

표 3. 메모리내의 포스팅 파일의 블록 구조
Table 3. Block structure of posting file in memory

number	dirty	Oid1	Oidn	next
--------	-------	------	-------	------	------

표3의 dirty 필드는 메모리로 적재된 블록이 갱신되었을 경우 디스크로 다시 반영을 할 것인지 아닌지를 결정하기 위한 필드이다. next 필드는 블록이 메모리에 적재되면 동적인 메모리 포인터 값을 할당받는다. 다음 [그림 3]에서는 디스크에서의 파일 구조와 메모리 적재후 변경된 이후의 파일 구조를 그림으로 도시하고 있다.

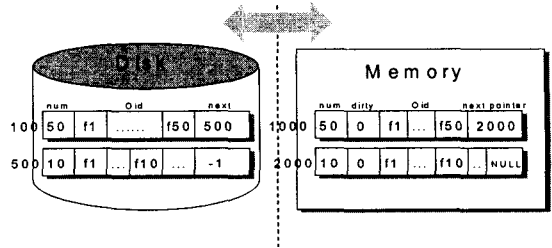


그림 3. 디스크와 메모리상의 포스팅 블록 변환
Fig 3. Posting block transformation between disk and memory

5. 문서와 이미지 내용을 이용한 문서 검색 방법

본 시스템에서 구현된 텍스트를 포함한 멀티미디어 데이터 이미지들 문서와 함께 검색하는 방법을 예로 설명하면 다음과 같다. 먼저, 사용자 인터페이스 단계로부터 boolean 연산을 이용한 질의를 "이미지의 내용이 Car인 멀티미디어 데이터를 포함한 문서 검색"과 다른 예로서 멀티미디어 데이터를 포함한 데이터에서 설명으로 이루어진 텍스트 중에 멀티미디어 데이터를 설명하는 텍스트에 대한 질의인 "Text에 Car가 포함된 문서 검색"의 질의를 입력받는다. [그림 4]에서는 멀티미디어 데이터로서 자동차 영상을 포함한 문서처리에 대한 예를 그림으로 도시하고 있다.

그림 4에서는 이미 입력된 자동차에 대한 문서를 검색할 경우 텍스트 인덱스 파일로부터 "Car"라는 키워드를 갖는 문서를 검색할 수 있으며, 또한 이미지의 캡션 정보의 인덱스 파일로부터 이미지의 내용이 자동차인 문서를 동시에 검색할 수 있다. 각각의 검색 결과들은 boolean 연산으로 and, or, not 등의 연산이 가능하고, 단어 가중치가 높은 문서부터 순차적으로 보여줄 수도 있다. 문서와 이미지의 내용이 다른 문서 검색도 검색이 가능하다.

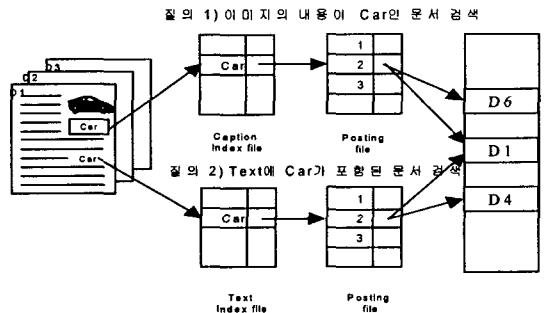


그림 4. 자동차에 대한 문서 검색
Fig 4. Document retrieval for car

검색 관리자에게 받아들여진 문서는 처음에 사용자가 하나의 파일을 저장하기 위해 사용하였던 것처럼 멀티미디어 데이터에 대한 내용을 기술한 문서와 같이 취급한다. 이렇게 질의 문의 문자열을 문서로서 받아들인 다음 다시 질의문상의 문서를 어휘분석기를 통하여 각각의 낱말들로 분류한다. 이들 낱말들은 아직 사용자가 원하는 파일을 찾기 위한 낱말 형식을 갖추고 있지 않다. 그러므로 각각의 낱말들에 대해 의미 없는 낱말들은 삭제한다. 이 부분에 해당되는 과정이 "stoplist" 과정이다. 이 과정을 거친 질의 문의 결과는 다시 "stemming" 과정을 거쳐서 실제로 데이터베이스 내의 인덱스 구조로 사용되는 B 트리를 검색하게 된다.

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.

IV. 결론 및 향후 연구

본 연구에서는 멀티미디어 데이터로서 영상 이미지를 포함한 문서에 대하여 텍스트 내용을 기반으로 한 문서의 검색 시스템을 설계하고 구현하였다. 이미지와 문서의 동시 접근을 위해 이미지는 캡션 정보를 이용하였고, 문서는 문서의 내용을 이용하여 접근하였다. 이는 기존의 문서 검색 방법이 이미지와 텍스트의 검색에 있어서 차별을 두는 반면에 같은 기법의 검색을 통해서 문서와 이미지를 동시에 검색을 수행할 수 있다. 또한 내용 기반의

이미지 검색보다 구현이 쉽고, 설명 정보가 정확하다면 검색의 정확도를 높일 수 있는 장점도 있다. 본 시스템의 특징으로 빠른 검색과 대용량의 문서를 위해서 B+트리를 이용한 역 파일 구조로 구현하여 동적으로 데이터의 블록을 생성함으로써 기억 장소의 낭비를 줄였다. 본 연구를 통한 응용 분야로써 영상 데이터 혹은 도면 등 텍스트를 포함한 멀티미디어 정보 관리나 특히 관련 정보 관리시 활용할 수 있고, 신문 및 잡지에 대한 멀티미디어 데이터 검색에도 활용할 수 있다. 향후 연구 과제로서 질의시 키워드의 부분 매칭이 가능한 시스템으로의 확장이 필요하고, 질의 모델을 확장하여 자연어를 지원하도록 하는 것이다. 또한 처리 속도에 따른 문제점이 존재하지만 멀티미디어 데이터인 이미지 자체 내용으로도 검색할 수 있도록 병행 연구가 필요하다.

참고문헌

- [1] Chung-Sheng Le, Rakesh Mohan and John R. Smith, "Multimedia Content Description In the InfoPyramid," IEEE ICASSP, Vol.6, pp3789-3792, 1998
- [2] Ajay divakaran, Hirofumi Nishikawa, Kohtarō Asai, "A Description Scheme For Video Based On Feature Extraction In the Compressed Domain", IEEE, pp278-279, 2000
- [3] William B. Frakes, Ricardo Baeza-Yates, "Information Retrieval : Data Structures & Algorithms", Prentice Hall Publishing
- [4] D. Tsichrizis and P. Economopoulos, "A Multimedia office filing system", Proc. of 9th Int. Conf. on VLDB, pp.2-7, 1983
- [5] Tzi-cker Chiueh, "Content-Based Image Indexing", Proceedings of the 20th VLDB Conference
- [6] 최기호, 문희정, 염성주, 김우생, 이원규, "내용을 기반으로한 이미지 검색 데이터베이스 시스템", 한국 정보 과학회지 제13권 제1호, pp.8-17 1995년 1월
- [7] 석상기, "멀티미디어 데이터베이스를 위한 객체지향 데이터 모델 및 검색 기법", 홍익대 박사 학위 논문, 1993
- [8] Virginia E. Ogle, Michael Stonebraker, "Chabot : Retrieval from a Relation Database of Images", IEEE Computer Vol. 28

No. 9, 1995

[9] Rohini K. Srihari, "Automatic Indexing and Content-Based Retrieval of Captioned Images", IEEE Computer Vol. 28 No. 9, 1995

[10] C. Faloutsos, "Access Methods for Text", ACM Computing Surveys, Vol. 17 No. 1, pp.49-74, March 1984

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 컴퓨터통계학과 교수
 ※관심분야 : 대규모 지식베이스, 인공지능경망, 이동에이전트

저자소개

이현창(Hyun-Chang Lee)



1993년 원광대학교 컴퓨터공학과 (공학사)
 1996년 홍익대학교 전자계산학과 (이학석사)
 2001년 홍익대학교 전자계산학과 (이학박사)

2001.3~2003.8 경인여자대학 전산정보학과 조교수
 2003.9~현재 한세대학교 컴퓨터공학과 조교수
 ※관심분야 : 멀티미디어 시스템, 주기억 데이터베이스 시스템, 웹 웨어하우징, XML