■ 2002년 정보과학 논문경진대회 수상작

문맥을 고려한 예제 기반 동영상 검색 알고리즘

(Content Based Video Retrieval by Example Considering Context)

박 주 현 † 낭 종 호 †† 김 경 수 ††† 하 명 환 ††† 정 병 희 †††

(Joo Hyoun Park) (Jong Ho Nang) (Gyung Su Kim) (Myung hwan Ha) (Byung Hee Jung)

요 약 효율적인 동영상 검색 방법은 많은 양의 동영상 데이터를 관리하는 디지털 비디오 라이브러리 시스템에서 필수적으로 요구되는 기능이다. 본 논문에서는 샷 단위 동영상을 문맥, 전경, 배경, 오디오로 나누어 비교하여 질의 동영상과 비슷한 동영상을 찾아내는 예제 기반 동영상 검색 알고리즘을 제안하였고, 제안한 알고리즘에 따라서 저작 및 검색도구를 구현하였다. 샷간의 관계 정보 즉, 문맥을 고려한다는 것은 인접한 샷들 간의 오디오, 움직임 정보들과 같은 저급 수준 내용 정보 간에 변화 패턴을 비교한다는 것이다. 두 번째 비교 요소인 전경은 움직이는 객체들의 집합을 의미하고, 세 번째 비교 요소인 배경은 전경을 제외한 나머지 비디오 정보를 의미한다. 이러한 비교 방법은 동영상 제작 과정에 근거한 것으로써 사용자로 하여금 직관적인 비교를 할 수 있게 한다. 또한 질의 신을 직접 구성할 수 있게 하였고, 각각의 비교요소에 가중치를 부여할 수 있도록 하여서 사용자의 검색의도를 자유롭게 반영할 수 있도록 하였다. 본 논문에서는 동영상이 가지고 있는 의미 정보를 검색에 완전히 반영하지는 못하지만, 문맥을 통해서 부분적인 의미 정보를 사용할 수 있도록 하였으며, 질의 신 구성과 직관적인 비교 요소를 사용함으로써 사용자의 검색 의도를 최대한 반영하고자 하였다.

키워드 : 내용 기반 동영상 검색, 예제 기반 검색

Abstract Digital Video Library System which manages a large amount of multimedia information requires efficient and effective retrieval methods. In this paper, we propose and implement a new video search and retrieval algorithm that compares the query video shot with the video shots in the archives in terms of foreground object, background image, audio, and its context. The foreground object is the region of the video image that has been changed in the successive frames of the shot, the background image is the remaining region of the video image, and the context is the relationship between the low-level features of the adjacent shots. Comparing these features is a result of reflecting the process of filming a moving picture, and it helps the user to submit a query focused on the desired features of the target video clips easily by adjusting their weights in the comparing process. Although the proposed search and retrieval algorithm could not totally reflect the high level semantics of the submitted query video, it tries to reflect the users' requirements as much as possible by considering the context of video clips and by adjusting its weight in the comparing process.

Key words : Content Based Video Retrieval, Query by Example

1. 서 론

컴퓨터 처리 능력의 향상과 인터넷과 같은 컴퓨터 네트워크의 빠른 성장은 컴퓨터에서의 정보 표현 수단을 텍스트 위주에서 동영상과 같은 멀티미디어 위주로 바꾸어 놓았다. 이미 인터넷 방송국과 같이 디지털 화된 동영상 데이타를 생산해내는 업체가 상당수에 이르며 기존의 아날로그 멀티미디어 데이타를 사용하던 방송국에서도 아날로그 데이타를 디지털 데이타로 변환하는 사업을 벌이고 있다. 이렇듯 기하급수적으로 제작되고 있는 멀티미디어 데이타, 그 중에서도 동영상 데이타를 효율적으로 관리하기 위해서 디지털 비디오 라이브러리 (Digital Video Library : DVL)가 구축되고 있으며, 저

† 비 회 원 : 서강대학교 컴퓨터학과
parkjh@mlneptune.sogang.ac.kr
†† 종신회원 : 서강대학교 컴퓨터학과 교수
jhnang@ccs.sogang.ac.kr
††† 비 회 원 : KBS기술연구소 연구원
odyssey.kbs.co.kr
: mhha@kbs.co.kr
bhjung@kbs.co.kr
논문접수 : 2002년 6월 21일
심사완료 : 2003년 8월 21일

장된 데이타가 많아짐에 따라 더욱더 강력한 동영상에 대한 내용 기반 검색 도구가 요구되어지고 있다.

동영상에 대한 내용기반 검색 방법으로서 텍스트 기반 검색과 예제 기반 검색을 생각해 볼 수 있다. 텍스트 기반 검색은 저작자가 동영상 데이타를 분석하여 텍스트로 표현한 후 변환된 텍스트 데이타를 검색에 이용하는 방법으로서 저작자의 의도를 가장 잘 반영할 수 있다는 장점이 있다. 반면에, 동영상의 구성 요소인 비디오와 오디오 데이타를 다른 미디어인 텍스트로 변환하는 과정에서 정보의 손실이 있을 수 있고, 저작하는 과정에 많은 노동력을 필요로 한다는 단점이 있다. 예제 기반 검색은 동영상 데이타를 직접 사용하기 때문에 정보의 손실이 없고, 질의 자체가 동영상으로 이루어지기 때문에 직관적인 검색이 가능하게 된다. 또한 저급 수준 정보를 자동으로 추출하여 사용하기 때문에 저작자의 노동력도 거의 들지 않는다는 장점이 있다. 하지만 현재의 기술 수준으로는 제한된 저급 수준 정보의 추출만이 가능하며, 추출된 저급 수준 정보를 가지고 동영상을 사람의 인지 수준으로 이해한다는 것은 불가능하기 때문에 고급 수준의 질의에 대한 처리를 완전히 수행하지 못한다는 단점이 있다. 하지만 제한된 저급 수준 내용 정보를 이용해서 사용자의 다양한 요구를 최대한 반영할 수 있는 방법이 필요하며, 본 논문에서는 직관적인 비교 요소 사용과 문맥의 사용을 제안함으로써 사용자의 다양한 요구를 수용할 수 있는 예제 기반 동영상 검색 알고리즘을 제안하고 시스템을 구현하였다. 일반적으로 동영상을 제작하는 방법은 배경 장소를 정한 다음에 사람이나 사물과 같은 객체를 배경 위에 위치시키고 카메라로 촬영하여 만들게 된다. 따라서 비교 단위를 색상이나 객체 단위가 아닌 전경(Foreground)과 배경(Background)으로 나누어 동영상 제작 방식에 근거하여 비교하는 것이 바람직하다. 또한 스토리를 가지는 동영상에 있어서 같은 신 내에 있는 샷들의 관계는 매우 큰 의미를 갖으며, 같은 샷이라도 전후 샷과의 관계에 의해 서로 다른 샷이 될 수 있다. 예를 들어 사람의 상반신이 나온 샷이 있다고 했을 때, 전후 샷의 상태에 따라서 대화 샷이 될 수도 있고, 단순한 줌인 샷이 될 수도 있다. 따라서 샷간의 관계를 동영상 검색에 이용하는 것은 좀 더 정확하고 다양한 요구를 반영할 수 있는 수단이 될 수 있을 것이다.

본 논문에서는 사용자의 다양한 요구를 현재의 기술 수준에서 최대한 반영할 수 있는 예제 기반 동영상 검색 알고리즘을 제안하고 이를 구현하였다. 예제 기반 검색 시스템의 궁극적인 목표는 저작자가 동영상을 직접 보고 이해한 내용을 기반으로 검색하는 시스템처럼, 시스템이 동영상의 내용을 자동으로 분석한 데이타를 사용하여 최대한 사용자의 의도를 반영할 수 있는 동영상 검색시스템을 구성하는 것이다.

2. 연구 배경

본 장에서는 예제 기반 동영상 검색을 이해하기 위해서 필요한 배경 지식에 관하여 설명한다. 2.1 절에서는 일반적인 예제 기반 동영상 검색 시스템(Query By Example 시스템: QBE 시스템)이 가지고 있는 동영상 검색 방법에 관해서 설명을 하고, 2.2 절에서는 본 연구에서 사용하고 있는 저급 수준 내용 정보들의 특징과 추출 방법에 관하여 설명한다. 2.3 절에서는 추출된 저급 수준 내용 정보들을 이용해서 고급 수준 정보인 객체를 추출하는 방법에 대해서 알아본다. 마지막으로 2.4, 2.5 절에서는 기존의 시스템들에 관하여 설명하고 문제점을 알아본다.

2.1 일반적인 예제 기반 동영상 검색 방법

일반적인 QBE 시스템에서 사용하는 동영상 검색 방법은 그림 1과 같이 표현할 수 있다. QBE 시스템은 크게 3가지 작업 단계로 분리할 수 있다. 첫 번째 단계는 선 처리 단계(Preprocessing)로서 비교 작업을 수행하는 데 있어서 필요한 저급 수준 정보들을 미리 추출하여 데이타베이스에 저장하는 단계이다. 이렇게 하는 이유는 사용자의 질의가 들어왔을 때 검색 대상이 되는 동영상의 저급 수준 내용 정보를 추출하게 된다면 연산 시간이 너무 많이 소요되기 때문이다. 두 번째 단계는 비교 단계(Matching)로서 질의로 들어온 동영상에 대해 선 처리 단계에서 했던 것과 같은 방법으로 저급 수준 내용 정보를 추출한 후 데이타베이스에 저장되어 있는 저급 수준 내용 정보와 비교를 하는 단계이다. 마지막 작업은 사용자 의견 반영 단계(Feedback)이다. 결과로 나온 동영상 집합에 대하여 사용자의 입력을 통해 가중치를 바꾸고 비교 단계를 다시 수행하게 된다. 이러한 예제 기반 동영상 검색 방법은 1995년 IBM의 QBIC 시스템[1]이 사용한 이후 대부분의 QBE 시스템에서 사용하고 있는 방법이다.

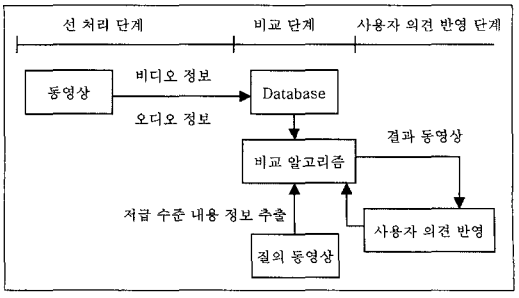


그림 1 일반적인 예제기반 동영상 검색 기법

2.2 저급 수준 내용 정보

동영상에서 추출할 수 있는 저급 수준 내용 정보는 크게 비디오 정보와 오디오 정보로 나눌 수 있다.

2.2.1 비디오 정보

MPEG-7에서는 멀티미디어 내용에 관한 기술 (Multimedia Contents Description)의 표준 안을 제안하고 있다. MPEG-7 관련 문서들에서 다루어지고 있는 비디오 정보[2]로는 색상(Color), 재질(Texture), 형태(Shape), 움직임(Motion) 등이 있으며, 본 연구에서도 이를 중심으로 하여 사용하였다.

2.2.2 오디오 정보

오디오 정보는 비디오 정보와 함께 동영상을 구성하는 주요한 특징이지만, 기존의 QBE시스템에서는 오디오 정보가 거의 사용되고 있지 않다. MPEG-7에서 오디오[3]는 저급 수준 정보 기술자(Descriptor)에 대한 표준안인 Audio Framework 부분과 Audio Framework의 기술자를 사용하여 만든 고급 수준 도구에 관한 부분인 High Level Tools 부분으로 나누어져 기술되어 있다. 본 연구에서는 MPEG-7에서 오디오에 관한 저급 수준 정보 기술자로 있는 Wave 타입 정보를 구성하고 있는 진폭(Amplitude)과 Fundamental Frequency 등을 사용하였다.

2.2.3 객체 추출

연속된 여러 프레임에 걸쳐 공간적, 시간적인 평가 척도 아래 연관된 영역들의 집합을 객체라고 할 수 있다. 객체를 추출하고 움직임을 따라가는 것은 매우 어려운 작업으로 많은 연구가 있어왔지만 아직까지도 완전히 배경과 객체를 분리해 내지는 못하고 있다. 본 연구에서는 S.F.Chang의 방법[4]을 기본으로 하여 객체를 추출하였다. 그림 2는 객체 추출 방법을 보여준다. 투사 및 색 영역 분리(Projection and Segmentation) 모듈은 현재 프레임의 공간적인 영역(Region)을 분리한다. 영역을 분리하기 위해서 이전 프레임의 움직임 정보를 사용한 투사(Projection)와 공간 적인 색 분리(Color Segmentation) 방법을 이용한다. 하지만 공간적인 영역 분리 방법만을 사용해서는 각 영역인 객체의 영역인지 배경의 영역인지를 구분하지 못한다. 이러한 이유로 시간적인 분리(Temporal Segmentation)작업이 이루어져야 하며 그 방법은 다음과 같다. 현재 프레임과 다음 프레임과의 사이에서 Optical Flow를 구하고, 이 정보를 바탕으로 나누어진 영역들의 움직임을 예측할 수 있다. 하나의 영역 안에 있는 모션 벡터들은 일정한 방향으로 하여 노이즈를 없애주고, 마지막으로 움직임이 일치하는 인접 영역들끼리 합쳐주어서 객체를 추출하게 된다.

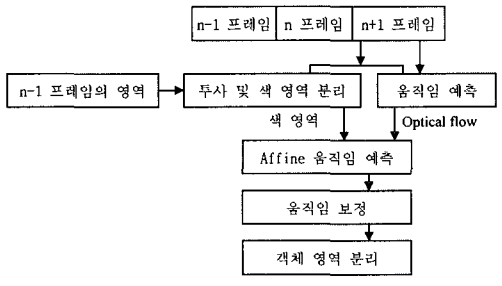


그림 2 객체 추출 방법

2.3 기존의 QBE 시스템과 문제점

초기의 QBE 시스템의 연구방향은 무엇을 가지고 비교를 해야 하는 가였다. 대표 프레임만을 선택해서 이미지 비교를 하는 방법[5]이나 객체를 추출해서 그 특성을 비교해서[6] 질의 동영상과 비슷한 동영상을 결정해야 한다는 연구들이 있었다. VideoQ 시스템[7]은 객체의 움직임을 중심으로 유사 동영상을 선택하였으며, 질의로 동영상이 아닌 움직임 스케치를 사용하였다. QBE 시스템의 가장 최근의 연구경향은 새로운 저급 수준 내용 정보 추출을 위한 연구보다는 비교 방법이나 검색 속도 향상 쪽에 초점이 맞추어져 있다. 비교 방법에 있어서 다항식 접근법[8]이나, 동적 프로그래밍 방법[9]을 사용한 시스템들은 기존의 여러 QBE 시스템들과 사용하는 저급 수준 내용 정보들은 같을지라도 비교 방법이나 데이타베이스 구성에 있어서 많은 차이를 나타내고 있다.

그러나 기존의 QBE 시스템들은 색상, 움직임, 형태, 재질, 객체, 오디오 정보 등에 대해 단순히 양적인 비교만을 통하여 비교 작업을 수행하고 있으며, 이러한 이유로 기존의 QBE 시스템을 동영상에 대한 내용 기반 검색이라고 하기에는 한계점을 가질 수밖에 없다. 동영상 데이타는 멀티미디어 데이타로서 비디오 미디어와 오디오 미디어를 동시에 포함하고 있다. 또한 사람이 동영상 데이타를 인지하는 과정에 있어서 시각정보뿐만이 아니라 청각 정보 또한 큰 비중을 차지함은 분명한 사실이다. 그럼에도 불구하고 기존의 QBE 시스템은 비디오 정보를 중심으로 하고 있으며, 오디오 정보를 사용한다고 해도 그 기여도가 그리 크지 않음을 알 수 있다. 표 1은 QBE 시스템들의 저급 수준 정보 사용 현황을 표로 나타낸 것이다.

표 1 기존의 QBE 시스템에서 사용한 저급 수준 정보

	색상	움직임	형태	재질	객체	오디오
동적 프로그래밍 방법	●	●	●			●
다항식 접근법 방법	●					
VideoQ	●	●	●	●	●	
대표 프레임 비교 방법	●					
객체 비교 방법	●	●	●	●	●	

3. 문맥을 고려한 예제 기반 동영상 검색 알고리즘

본 장에서는 예제 기반 동영상 검색을 하기 위한 비교 요소의 표현 방법과 비교 방법, 검색 알고리즘에 대하여 설명한다. 3.1절에서는 저급 수준 내용 정보의 표현과 비교 방법에 대해서 알아보고, 3.2 절에서는 문맥의 표현과 비교 방법에 대하여 설명한다. 마지막으로 3.3 절에서는 문맥을 고려한 예제 기반 동영상 검색 알고리즘을 제안한다.

3.1 저급 수준 내용 정보의 표현과 비교 방법

본 연구에서는 저급 수준 내용 정보를 전경과 배경으로 나누어서 사용한다. 영화를 찍을 때 카메라에서 가까운 곳은 전경이고 카메라에서 먼 곳은 배경이 된다. 일반적으로 영화의 제작 과정은 먼저 배경을 설정하고 그 배경 위에 등장 인물전경-을 배치시킴으로써 이루어진다. 따라서 모든 영화의 한 장면은 전경과 배경으로 나눌 수 있으며 이는 영화뿐만이 아니라 일반적인 동영상에까지 확대하여 적용할 수 있다. 이러한 관점에서 보면, 전경과 배경은 직접적인 연관성이 없으므로 서로 분리하여 독립적인 비교 요소로서 사용하는 것은 타당성을 갖으며 사용자가 원하는 검색 종류를 직관적으로 선택할 수 있게 해줄 수 있다.

3.1.1 전경

저급 수준 내용 정보의 분석 측면에서 보았을 때, 전경은 객체들의 집합으로 정의 할 수 있다. S.F.Chang의 방법[4]을 사용하여 추출된 객체는 색상, 형태, 크기, 움직임 등의 특징을 가지고 구성되어 있으며 이러한 객체의 저급 수준 내용 정보들은 전경에 대한 기본적인 비교요소로서 사용되어 진다. 기본적인 비교 단위로 샷을 사용하였을 때 객체의 저급 수준 내용 정보들은 크게 변할 수도 있지만 일반적으로는 일정 범위 안에서 유지된다. 따라서 모든 프레임에 대해 객체의 저급 수준 내용 정보를 비교하는 것은 매우 큰 연산 시간을 요구하며 비교 성능에 있어서도 의미를 찾기가 힘들다. 따라서 본 연구에서는 움직임 정보나 위치 정보와 같은 변화가 의미를 갖는 저급 수준 정보를 제외하고는 샷 안에서 가장 오래 유지되는 객체를 대표 객체로 하여 비교 분석에 사용한다. 본 연구에서 전경 객체들의 저급 수준 내용 정보로서 사용한 것은 색상 히스토그램, 형태, 위치, 움직임, 영역 등이며 이는 MPEG-7 Visual부분[2]에서 동영상 비디오의 내용 정보에 관한 기술자로서 사용되고 있는 것들이다.

• Luminance 히스토그램

LUV 색상 포맷은 이미지 분석에 있어서 가장 많이 사용되는 포맷이다. Luminance는 단순한 밝기 정보가 아니라 사람에 눈에 인식되기 쉽도록 녹색 영역에 더 많은 가중치를 부여한 밝기 정보이다. Luminance의 변화만 사용해서도 샷 경계 검출[10]이나 움직임을 예측할 수 있을 정도로 중요한 이미지 내용 정보를 제공하고 있다. 본 연구에서도 객체간의 밝기 톤을 비교하기 위해서 Luminance 히스토그램을 사용하였으며 비유사도는 다음과 같이 산출하였다. j개의 bin을 가진 질의 동영상의 Luminance 히스토그램을 HQ_j라 하고, 대상 동영상의 히스토그램을 HD_j라 한다면 비유사도 DS_0^F는 히스토그램 교차방법(Histogram Intersection)[1]을 사용하여 식 (1)과 같이 구할 수 있다.

$$DS_0^F = 1 - \frac{\sum_j \min HQ_j HD_j}{\sum_j HQ_j} \quad (1)$$

• 객체 영역 색상

Q의 모든 영역에 대하여 객체의 중심에 대한 영역의 상대적인 위치가 비슷한 D의 영역을 찾아 Euclidean Distance를 구하게 된다. 구한 비유사도는 정규화 과정을 거친 후 사용한다. 객체를 추출하기 위해선 먼저 영역을 추출해야 한다. 영역은 색 분리와 Optical Flow를 통해서 구할 수 있으며, 움직임이 동일한 영역을 하나로 묶어서 객체를 구성하기 때문에 객체 추출의 기본 단계로서 사용된다. 색 정보에 대해서는 이미 Luminance 히스토그램 비교 방법을 사용하여 비교하였지만 이는 객체 전체의 밝기 톤에 대한 비교를 할 수 있을 뿐이지 구체적인 색상 분포는 비교할 수가 없다는 문제점을 가지고 있다. 따라서 본 연구에서는 색 분리 결과로 나온 영역에 대한 색상 비교를 수행함으로써 Luminance 색상 비교에 대한 문제점을 보완하였다. 색상 비교는 LUV 또는 RGB 형식을 사용할 수 있으며, 본 연구에서는 LUV비교를 수행하였다. 질의 동영상의 객체 Q와 대상 객체 D에 대한 영역별 색 비교를 한다고 하자. 객체 Q에 영역이 m개 있고, D에 n개의 영역이 있다고 했을 때, 먼저 각 영역별로 샷 내의 모든 프레임에 대해서 평균 색상을 구하여 영역에 대한 대표 색상으로 설정한다. 객체 Q의 영역 i에 대한 대표 색상을 $C_{qi}(L, U, V)$ $(0 \le i \le m-1)$라 표현하고, 객체 D의 영역 j에 대한 대표 색상을 $C_{dj}(L, U, V)$ $(0 \le j \le n-1)$라 하자. 그리고 Q 객체의 중심 좌표를 (x_q, y_q)라 하고, D 객체의 중심을 (x_d, y_d)라 했을 때, 영역들의 중심을 각각 (x_{qi}, y_{qi}), (x_{dj}, y_{dj})라 한다면, 비유사도 DS_1^F는 Euclidean Distance를 이용해서 식 (2)와 같이 구할 수 있다. Q의 모든 영역에 대하여 객체의 중심에 대한 영역의 상대적인 위치가 비슷한 D의 영역을 찾아 DS_1^F을 구하게 되며, 정규화 과정을 거친 후 사용한다.

$$DS_1^F = \sum_i \sqrt{(c_{qi}(L)-c_{dj}(L))^2+(c_{qi}(U)-c_{dj}(U))^2+(c_{qi}(V)-c_{dj}(V))^2}$$

$$where,\ \arg\min_{i,j}\{|(x_{qi}-x_{dj})-(x_q-x_d), y_{qi}-y_{dj})-(y_q-y_d))|\} \quad (2)$$

• 형태

형태를 비교하는 데는 비트맵 마스크 (Bitmap Mask)가 사용된다. 비트맵 마스크를 사용하여 비교를 하기 때문에 프레임 크기가 틀리다면 먼저 프레임 사이즈를 맞추어 주어야 하고 비트맵의 가운데로 객체를 평행 이동시켜 비교하여야 한다. 그리고 전체에 대한 교차 부분(Intersection) 정도의 산출을 통해 비유사도를 구할 수 있다. 질의 비트맵 마스크를 A라 하고 대상 비트맵 마스크를 B라 했을 때, 가로 i번째, 세로 j번째 블록이 객체에 포함되면 $A_{ij}=1$, 포함되지 않으면 $A_{ij}=0$ 이라 한다면, A,B의 비트맵 마스크의 비유사도 DS_2^F는 식 (3)과 같이 표현할 수 있다.

$$DS_2^F = 1 - \frac{\sum_{ij} A_{ij}B_{ij}}{\sum_{ij} A_{ij} + \sum_{ij} B_{ij} - \sum_{ij} A_{ij}B_{ij}} \quad (3)$$

• 위치

객체의 위치는 프레임 안에서 객체 중심의 좌표를 의미한다. 하나의 샷 안에서 객체의 위치는 변할 수 있지만 그 변화량과 변화정도는 움직임 정보를 비교할 때에 다루어지기 때문에 여기서는 무시하도록 한다. 대신 샷 안에서 가장 오랫동안 유지되는 위치를 객체의 위치로 간주하고 Euclidean Distance를 사용해서 비교한다. (x_q, y_q)를 질의 동영상의 객체의 위치라 하고, (x_d, y_d)를 대상 객체의 위치라고 한다면 비유사도 DS_3^F는 식 (4)과 같이 구할 수 있으며, 프레임 사이즈로 정규화 하여 사용한다.

$$DS_3^F = \sqrt{(x_q-x_d)^2+(y_q-y_d)^2} \quad (4)$$

• 움직임

객체 기반의 QBE 시스템[6][7]에서는 움직임에 대한 비교를 매우 중요하게 다루고 있다. VideoQ[7]에서는 객체의 움직임 정보를 각각의 프레임에 대해 객체 움직임에 대한 벡터를 구성하여 비교하고 있다. 이러한 비교는 움직임에 매우 민감한 검색을 할 수 있다는 장점을 가지고 있지만 모든 프레임에 대한 비교를 해야 하기 때문에 계산량이 많아진다는 단점이 있다. M.R.Naphade의 시스템[9]에서는 움직임의 크기와 방향을 히스토그램으로 만들어 비교하는 방법을 사용하고 있다. 이 방법에서는 전체 프레임에 대한 움직임 히스토그램을 만들어 사용하기 때문에 전체적인 움직임 에너지와 움직임 방향을 비교할 수 있다는 장점을 가지고 있는 반면에 부분적인 정보가 손실 될 수 있다는 단점이 있으며, 역시 모든 프레임에 대해서 각각 히스토그램을 만들어 비교하기 때문에 계산 량이 많아진다는 단점을 가지고 있다. 본 연구에서는 두 방법을 혼합하여 사용하였다. 프레임 전체가 아닌 각각의 객체에 대해서 움직임 크기, 방향 히스토그램을 만든 후에 각 빈들에 대한 평균값을 산출하여 교차 정도를 측정하는 방법을 사용하였다. 이러한 방법을 사용하게 되면, 샷 안에서 해당 객체의 움직임 정도와 편향되게 움직였다면 그 움직임 방향 또한 알 수 있지만, 움직임 방향이 일관되지 않다면 방향 정보가 사라질 가능성이 존재하며 시간 정보를 아예 표현할 수 없다는 단점을 가지고 있다. 무엇보다도 이 방법을 사용하는 가장 큰 장점은 평균값을 사용하여 비교하기 때문에 비교 시간 면에 있어서 다른 어떤 방법보다도 빠르다는 것이다. 움직임 크기와 방향 히스토그램의 추출과 비교 방법은 다음과 같다. 움직임 크기와 방향은 블록 단위로 구한 Optical Flow를 사용하여 구한다. 각 블록에서 Optical Flow가 (d_x, d_y)벡터로 구해진다고 하면, 움직임 크기는 식 (5), 방향은 식 (6)과 같이 구할 수 있다.

$$m = \sqrt{d_x^2+d_y^2} \quad (5)$$

$$\theta = \arctan\left(\frac{d_y}{d_x}\right) \quad (6)$$

움직임 크기에 대한 비유사도 DS_4^F와 움직임 방향에 대한 비유사도 DS_5^F는 식 (1)과 같이 히스토그램에 대한 비교 방법으로 많이 쓰이는 교차 비교 방법을 사용하였다.

Luminance 히스토그램, 영역, 색상, 형태, 위치, 움직임 크기 및 방향 히스토그램을 각각 비교하여 산출된 비유사도를 정규화 한 뒤 합산하여 전경에 대한 전체 비유사도로 결정한다. 따라서 전경에 대한 전체 비유사도 DS^F는 식 (7)과 같이 구할 수 있다. w_i는 각각의 비교요소들에 대한 가중치를 의미하며, DS_i^F는 정규 화된 비교요소들의 비유사도를 의미한다.

$$DS^F = \sum_{i=0}^{5} w_i DS_i^F \quad (7)$$

3.1.2 배경

배경은 전경으로 분류된 객체들을 제외한 나머지 영역들을 의미한다. 전경과 달리 배경은 카메라의 움직임의 여부에 따라 비교 방법이 틀려지게 된다. 샷 안에서 카메라의 움직임이 없다면 배경은 변하지 않으며, 카메라의 움직임이 있으면 배경은 카메라의 움직임에 따라 변하게 된다. 동영상에서 카메라 오퍼레이션(Camera Operation)을 감지하고 그 종류까지 추출해내는 방법은 이미 많은 연구가 이루어져 왔지만, 정확성과 소요시간 면에서 아직까지도 좋은 성능을 내고 있지는 못하다. 본 연구에서는 카메라 움직임의 종류보다는 카메라의 움직임 여부만을 필요로 하기 때문에 간단히 Optical Flow

와 블록 비교를 통해서 정보를 얻어낼 수 있다. Optical Flow가 프레임 전체에 걸쳐 일정 크기 이상의 나타내는 경우에는 카메라 움직임이 있을 확률이 크고, 객체가 위치하지 않을 확률이 제일 높은 프레임 상단의 블록들을 다음 프레임의 블록과 비교함으로써 카메라의 움직임 여부 확률을 계산할 수 있다[11].

카메라의 움직임이 없을 경우의 배경 비교는 다음과 같은 방법으로 진행한다. 카메라의 움직임이 없을 경우의 배경은 움직이는 객체에 의해 가려지거나 드러나서 변하는 부분 외에는 크게 변하지 않는다. 따라서 Dissolve나 Fade-In, Fade-Out과 같은 샷 전이 효과를 배제할 수 있는 중간 프레임 배경 하나만을 비교 프레임으로 선정한다. 카메라의 움직임이 있을 경우에는 비교 프레임을 여러 개 선정하여 사용한다. 비교 프레임 선정 작업은 객체를 제외한 영역에 대한 히스토그램을 구하여 선정할 수 있다. 먼저 각 빈에 대해 평균을 구한 후에 평균과 교차가 많은 프레임과 적은 프레임을 선정하여 비교 프레임으로서 사용한다. 선정 작업은 평균과 가까운 정도에 대한 단계를 두어 비교 프레임 개수를 조절하여 사용한다. 비교 프레임이 많아지면 비교 작업에 있어서 정확도는 높아지지만 계산량이 많아지게 된다. 각 샷에서 선택된 한 개 이상의 대표 프레임들에 대한 비교 작업은 이미지에 대한 비교작업으로 생각할 수 있다. 본 연구에서는 이미지 비교 알고리즘인 Blob 히스토그램 방법[12]을 사용하여 배경 프레임을 비교하였다. 질의 동영상의 대표 배경 이미지 n장을 I_{qi}라 하고, 대상 동영상의 대표 배경 이미지 m장을 I_{dj}라 한다면, 각각의 이미지에 대해서 가능한 모든 조합에 대해 비교를 수행한 후에 비 유사도가 가장 낮은 값을 배경에 대한 비유사도로서 결정한다.

$$DS^B = \min_{0<i<n,\,0<j<m} (BlobHistogramMatching(I_{qi}, I_{dj})) \quad (8)$$

3.1.3 오디오 정보

동영상에 자동으로 의미 정보를 추출하는 것이 주된 목적이었던 MoCA(Automatic Movie Content Analysis) 프로젝트에서는 비디오 정보와 더불어 오디오 정보에서도 의미 정보를 추출하기 위한 연구를 진행하였다. MoCA 프로젝트의 오디오 부분에 관한 연구[13]에서는 오디오에 대한 기본 성질에 관한 설명에서 오디오를 물리적인 면(Physical Properties)과 심리적인 면(Phyco-Acoustical Properties)으로 분류하여 응용하였다. 본 연구에서는 오디오의 내용분석이 목적이 아니라 두 오디오의 유사도 측정이 목적이기 때문에 오디오의 물리적인 면 중에서 소리의 크기를 나타내는 진폭(Amplitude)과 음색의 특색을 나타내는 주파수(Frequency)만을 비교 대상으로 사용하였다.

• 진폭과 주파수

두 동영상에 포함된 오디오의 진폭을 비교하는데 있어서 절대적인 비교를 하는 것은 큰 의미가 없다. 진폭은 오디오의 볼륨과 직결되는 성질이기 때문에 오디오의 녹음 상태에 큰 영향을 받기 때문이다. 따라서 진폭에 대해서는 변화의 패턴이 중요한 비교 요소가 되고, 이는 샷 단위의 변화 패턴을 비교하는 문맥 비교에서 사용한다. 샷 안에서의 진폭에 대한 비교는 진폭에 대한 RMS평균과 분산을 구해 질의 동영상과 대상 동영상의 진폭 차와 분포만 비교한다. 각각에 대해 Euclidean Distance를 구해 합하여 DS_0^A로 정의한다.

FFT(Fast Fourier Transform)를 통해 산출한 각 주파수 영역대의 계수 값으로 히스토그램을 만들어 <식 1>과 같은 교차점 방식을 통해 비교한다. 이러한 비교 방법을 사용하면 가장 강조된 주파수 영역대가 일치하는 지의 여부를 알 수 있다. 주파수대에 대한 비유사도는 DS_1^A 로 표현한다.

오디오에 대한 전체 비유사도 DS^A는 식 (9)와 같이 구할 수 있다.

$$DS^A = \sum_{i=0}^{1} w_i DS_i^A \quad (9)$$

3.2 문맥 정보의 표현과 비교 방법

문맥의 사전적 의미는 문장의 각 성분 사이에 성립하는 의미론적이며 논리적인 관계를 총칭하는 것이다. 일반적으로 문맥은 자연어와 관련된 용어로서 통용되고 있다. 컴퓨터 과학의 한 분야인 자연어 처리 분야에서 단어의 중의성을 해결하기 위해서 단어의 의미를 문장 안에서 파악하는 문맥이라는 개념이 사용되고 있다. 예를 들어 "나는 배가 아프다."라는 문장과 "나는 배를 탔다."라는 두개의 문장을 생각해보자. "배"라는 단어는 같은 형태를 가지고 있지만, 두 문장에서 서로 다른 뜻으로 사용되고 있다. "배"라는 단어는 문장 안에서 해석할 때에만 올바르게 해석할 수 있다. 문맥의 개념을 조금 확장해보면 스토리를 가진 동영상에도 적용해볼 수 있다. 동영상에서의 샷을 자연어에서의 단어로 본다면 샷이 가지고 있는 중의성을 신 안의 다른 샷들에 대한 정보를 통해 해결할 수 있을 것이다. 즉, 샷은 독자적으로 의미를 가질 수도 있지만, 주위의 샷들의 영향을 받아 그 의미가 더 분명해 질 수 있는 것이다. 실제로 영화 연출론 서적인 Shot By Shot[14]에서 문맥이란 단어를 사용하고 있으며 3개의 샷이 있을 때, 샷의 배치에 따라서 의미정보가 바뀔 수 있다는 것을 예를 통해서 설명하고 있다.

스토리를 가진 동영상에서의 문맥 개념을 실제 QBE 시스템에 적용하여 사용할 수 있다. 예를 들어 사용자가 대화 샷을 찾기 위해서 사람의 상반신이 나온 샷을 질

의 샷으로 주었을 때 문맥을 고려하지 않는다면 사람의 상반신이 나온 모든 샷을 검색해낼 것이다. 하지만 문맥을 적용하게 된다면 대화 샷이 가진 반복 패턴에 의해 대화 중에 있는 사람의 상반신 샷만을 검색해낼 수 있다. 문맥을 사용하게 된다면 샷간에 저급 수준 내용 정보에 대한 관계를 고려함으로써 기존의 방법보다는 조금 더 사용자의 의도에 맞는 예제 기반 동영상 검색을 할 수 있을 것이다.

3.2.1 문맥 정보 표현에 사용되는 저급 수준 내용 정보

문맥 정보는 신 단위로 추출하여 사용한다. 스토리를 가진 동영상에 있어서 신은 동일한 장소와 동일한 인물로 구성되는 것이 보통이기 때문에 저급 수준 내용 정보들의 변화 패턴은 중요한 의미를 갖는다. 다음은 샷간의 관계를 고려할 수 있는 저급 수준 내용 정보와 그 특징 및 사용 이유에 관하여 설명한 것이다.

• Luminance

Luminance 문맥 정보는 샷 간 밝기 톤에 대한 변화 패턴을 비교할 수 있다. 대개의 경우 하나의 신 안에서 샷 간 Luminance 값은 샷 변화 지점에 있어서 어느 정도 변화를 보일 수 있지만 큰 폭의 변화는 조명의 변화이거나 폭발 장면과 같은 특정한 이벤트가 벌어졌음을 의미할 수 있다.

• 움직임 크기

샷 간 움직임의 큰 변화는 카메라 움직임의 폭의 변화가 커지거나 객체의 움직임의 변화가 큰 경우에 나타날 수 있다. 예를 들어 움직임의 크기가 작다가 갑자기 큰 폭으로 커진다면 새로운 이벤트의 도입부일 가능성이 매우 높아지게 된다. 또한 움직임의 크기가 크다가 작아진다면 이벤트의 소강상태일 가능성이 높다는 것을 의미한다. 이렇듯 움직임의 크기 변화 패턴 또한 동영상에서 문맥을 결정하는데 중요한 의미를 가질 수 있다.

• 진폭, 주파수

오디오 정보만을 사용한 신 경계 검출[15]이 가능할 정도로 오디오 정보는 신 단위 분석 시에 매우 유용하게 쓰일 수 있다. 예를 들어, 하나의 신 안에서는 진폭이나 주파수의 변화 폭이 작게 유지되는 경우가 일반적이므로 진폭이나 주파수의 큰 변화 패턴을 살펴보면 폭발이나 액션, 공포, 전쟁 장면처럼 매우 큰 이벤트의 발생이나 소멸을 찾아낼 수 있다.

• 샷 길이

샷 길이의 패턴 또한 중요한 문맥 요소로서 사용 될 수 있다. 같은 길이를 가진 샷이라도 전 후 샷들의 길이에 따라 짧은 샷이 될 수도 있고, 긴 샷이 될 수도 있기 때문이다.

• 샷 유사도

저급 수준 내용 정보들을 같은 신 안에 있는 샷들과 비교하여 샷 반복 패턴을 만들어 사용할 수 있다. 대화 샷과 같은 특별한 샷 반복 패턴을 가지고 있는 샷 검색에 유용하게 사용될 수 있다.

3.2.1 표현 및 비교 방법

샷 유사도를 제외한 4개의 문맥 비교 요소는 평균과 분산 값을 이용하여 contour로 구성하여 표현한다. Contour에 의한 비교는 음악분야의 내용 기반 검색[16]에서 사용하는 방법으로 시퀀스와 높낮이를 가지고 있는 음표에 대한 비교를 하기에 적절한 방법이다. 음악분야에서 사용하는 contour는 바로 전 음표와의 높이비교에 따라 현재 음표의 contour가 결정되게 된다. 문맥 정보들도 신 안에서의 평균에 비해 높은 샷이 있고 낮은 샷이 있으며 높낮이 시퀀스를 형성하고 있기 때문에 contour에 의한 비교 방법을 적용할 수 있다. 하지만 각 샷들의 contour를 결정하는 데 있어서 바로 앞 샷의 정보만 보고 결정한다면 신 안에서의 현재 샷의 의미를 잃어버릴 수가 있다. 따라서 본 연구에서는 contour의 표현 방법을 기존의 방법에서 약간 응용하여 신 전체에서의 샷의 위치를 표현할 수 있도록 다음과 같이 변형하여 사용하였다.

4가지 비교요소에 대한 contour 표현 방법은 다음과 같다. 먼저 신 안의 샷들의 평균값을 사용하여 신 평균을 구한다. 신 평균에 비하여 샷의 평균값이 임계 값보다 높은 값을 가지고 있으면 H로 표현을 하고, 임계 값보다 낮으면 L, 임계 값보다 높거나 낮지 않으면 E로 표현을 한다고 하자. 임계 값은 각 샷의 분산 값에 의해서 적절하게 조정하여 사용할 수 있다. 샷의 분산 값이 높게 되면 샷 안에서의 변화가 큰 것이기 때문에 평균값이 분산이 낮은 샷에 비해서 큰 의미를 가질 수 없다. 이러한 경우에는 임계 값을 상향조정하여 설정한다. 예를 들어 폭발 신의 contour를 분석해보면, 대부분의 경우 폭발이 시작되기 전 긴장감을 표현하기 위해서 오디오의 진폭은 평균보다 작으며, 움직임, Luminance 모두 평균보다 작은 값을 갖게 된다. 따라서 폭발 신의 시작은 대개의 경우 L로 시작을 한다. 잠시 후 폭발이 일어나게 되고 폭발이 일어나는 샷부터 오디오의 진폭, Luminance, 움직임 등이 모두 H값을 가지게 된다. 비단 폭발 신만이 아니라, 다른 일반적인 신들도 저급 수준 정보간에 일정한 패턴을 가지고 있으며 이것은 샷 분석에 있어서 중요한 단서가 될 수 있다.

Contour의 비교는 문자열 비교를 통해서 간단히 할 수 있으며, 오류 허용 값을 설정해 비교의 정확도 정도를 조정한다. 예를 들어 질의 동영상의 샷 패턴이 HLL이라 하고 가운데 샷이 질의 샷이고 처음과 마지막 샷이 문맥 도움 샷이라고 하자. 오류 허용 값이 0이면 정확히 일치하는 패턴을 데이타베이스에서 검색하게 되고,

오류 허용 값이 1이면 LLL, ELL, HHL, HEL, HLH, HLE, HLL과 일치되는 패턴을 데이타베이스에서 검색하게 된다. 이러한 경우 비슷한 샷으로 보이는 결과 샷은 가운데 샷이 된다. 질의 동영상의 샷들을 S_{qi}, 샷 패턴 비교를 통해 선정된 샷들을 S_{di}라 하고 샷의 개수를 N이라 하였을 때, 검색된 샷에 대한 문맥 비유사도 DS^c는 식 (10)과 같이 계산할 수 있다.

$$DS^c = \sum_{j}^{\text{문맥비교요소}} w_j \sum_{i}^{N} |(m^j(S_{qi}) - m_q^j) - (m^j(S_{di}) - m_d^j)|(error\ term_j + 1) \quad (10)$$

m_q^j, m_d^j는 각각 질의 신과 대상 신의 각 문맥 요소들에 대한 평균을, $m^j(*)$는 각 샷들의 평균을 의미하며 *error term*은 일치하지 않는 contour의 수를 의미한다.

3.3 문맥을 고려한 예제 기반 동영상 검색

전경, 배경, 오디오, 그리고 문맥 정보를 사용하여 문맥을 고려한 예제 기반 동영상 검색을 할 수 있으며 전체 비유사도는 식 (11)과 같이 계산할 수 있다. 임계 값 이하의 값을 갖는 DS가 낮은 샷부터 순위를 주어 일정 개수의 샷을 유사한 샷으로 결정할 수 있다.

$$DS = w^F DS^F + w^B DS^B + w^A DS^A + w^C DS^C \quad (11)$$

그림 3은 동영상 검색 과정을 간단하게 보여준다. 동영상 검색을 하기 위해서는 먼저 비디오에서 저급 수준 내용 정보들을 추출하여 데이타베이스에 저장해 두어야 한다. 일반적으로 저급 수준 내용 정보를 추출하는 데에는 많은 시간이 소요되므로 미리 데이타베이스를 구성해두는 것이다. 문맥 정보를 제외한 저급 수준 내용 정보들은 샷 단위로 저장이 되며 샷 단위 추출 정보를 기반으로 구성되는 문맥 정보는 신단위로 저장한다. 이러한 작업을 담당하는 모듈이 각각 저급수준 내용정보 추출모듈과 신 단위 문맥 contour 생성 모듈이다. 질의 동영상이 들어오게 되면 같은 방법으로 저급 수준 내용 정보와 문맥 정보를 추출한 후 비교 작업을 통해 비유

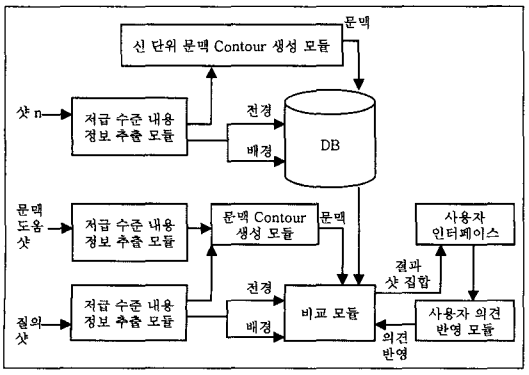


그림 3 문맥을 고려한 예제기반 동영상 검색 과정

```
Procedure Retrieval_With_Context(query_scene)
Begin
    query_shot <- GetShot(query_scene);    // 질의 샷을 얻는다.
    while(TRUE) do
        dest_shot <- GetShotFromDB();    // DB에서 샷을 얻는다.
        if(dest_shot = NULL) // DB의 모든 샷을 얻었으면 비교를 마친다.
          return ResultShotSet;
        endif
        dest_scene <- GetSceneFromDB(); // dest_shot이 포함되어 있는 신을 얻는다.
        //각 비교요소에 대한 비유사도를 구하여 정규화한다.
        DS_F <- Norm( CompareForground(query_shot, dest_shot) );
        DS_B <- Norm( CompareBackground(query_shot,dest_shot) );
        DS_A <- Norm( CompareAudio(query_shot, dest_shot) );
        DS_C <-Norm( CompareContext(query_scene, dest_scene) );
        //전체 비유사도를 구한다.
        DS <- (W_F*DS_F+ W_B*DS_B+ W_A*DS_A+ W_C*DS_C);
        if(isCheckShotSimilarityPattern)//사용자가 패턴비교를 선택하면
            DS <- DS + SamePatternCheckShot(query_scene, dest_scene);
            //패턴이 일치 하지 않은 정도에 따라 비유사도를 더한다.
        endif
        //현재 결과샷들의 비유사도보다 낮으면 결과 샷에 넣고 비유사도 최고값을 갱신.
        if(DS < MAX_DS)
          MAX_DS<-ResultShotSet(ResultShotSet,dest_shot, DS);
        endif
    endwhile
end
```

그림 4 문맥을 고려한 예제기반 동영상 검색 알고리즘

사도를 구하게 되며, 이와 같은 작업을 하는 모듈이 비교 모듈이다. 비교 작업은 전경, 배경, 오디오, 문맥에 대하여 비유사도를 구한 후에 사용자가 입력한 가중치를 적용하여 전체 비유사도를 구하게 된다. 전체 비유사도가 임계 값보다 작다면 후보 샷으로 선정한 후, 후보 샷들에 대해서 질의 동영상의 샷 유사도 패턴과 일치하는지의 여부를 조사한다. 샷 유사도 패턴까지 일치한다면 대상 샷은 질의 샷과 유사한 샷으로 선정되게 된다. 비교 작업은 비교 모듈에서 담당하며 그림 4는 문맥을 고려한 예제기반 동영상 검색 알고리즘을 의사 코드로 작성한 것이다.

4. 시스템 구현 및 실험 결과 분석

제안한 알고리즘을 기반으로 하여 본 연구의 최종 결과인 동영상 검색 시스템을 구현하였으며 몇 개의 동영상으로 데이타베이스를 구성하여 실험을 하였다. 본 장에서는 시스템 구현 방법과 사용자 인터페이스 및 실험 방법에 대하여 설명을 하고 실험 결과를 분석한다.

4.1 문맥을 고려한 예제기반 동영상 검색 시스템 구현

일반적으로 검색 시스템은 클라이언트/서버 구조로서 구성된다. 본 연구를 위해서 구현한 시스템도 이와 같은 구조로 구성되어 있으며, 장기적으로 볼 때 저작도구와 검색도구를 분리하여 검색도구는 웹 클라이언트로 구현하는 것이 바람직할 것이다. 그림 5는 전체적인 시스템 구조를 보여준다. 동영상 검색을 하기 위해서 가장 먼저 해야 할 일은 동영상에서 필요한 정보를 추출해내는 것이다. 이러한 작업은 그 소요시간이 많이 걸리기 때문에

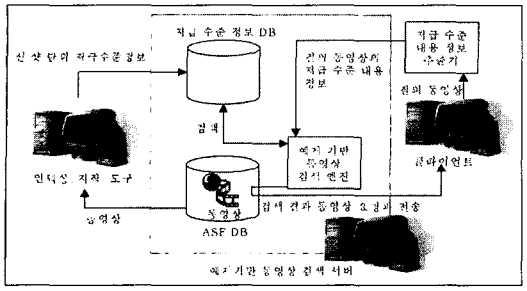


그림 5 문맥을 고려한 예제기반 동영상 검색 시스템 구조

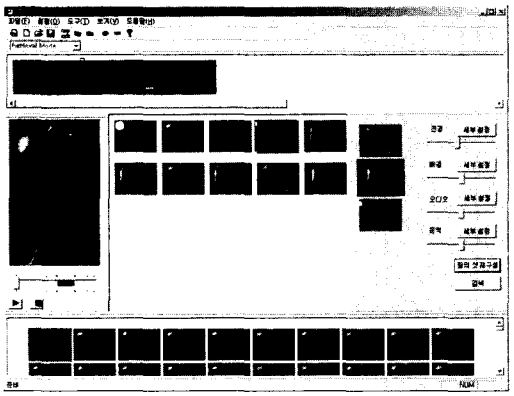


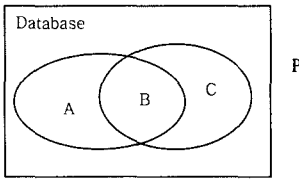
그림 6 저작 및 검색도구와 질의 동영상 구성

일반적으로 오프라인 상에서 이루어지며 동영상을 샷과 신으로 나누어 검색에 필요한 저급 수준 내용 정보들을 추출하여 데이타베이스에 저장하는 역할을 한다. 동영상 검색은 전용 클라이언트 프로그램을 구현하여 사용하였는데, 이는 사용자가 질의 샷을 선택하고 문맥 도움 샷을 선택하면 이들로부터 전경, 배경, 오디오, 문맥 정보를 추출한 후 이 정보들을 서버로 넘겨주는 일을 한다. 질의 동영상의 저급 수준 내용 정보를 넘겨받은 서버는 검색 엔진을 통해서 미리 추출해 둔 대상 동영상들의 저급 수준 내용 정보들과 비교한 후 비유사도가 가장 낮은 샷부터 순위를 매겨 비슷한 동영상을 찾아내게 된다. 서버는 결과 동영상들을 다시 클라이언트로 전송해 주고 클라이언트는 전송 받은 결과 동영상을 사용자에게 보여줌으로써 검색을 마치게 된다.

같은 질의 동영상이라도 검색하는 사람에 따라서 그 의도가 틀릴 수 있을 것이다. 예를 들어 바닷가에 서 있는 높은 건물이 폭파되는 신의 한 샷을 질의 샷으로 했을 때, 어떤 사람은 바닷가에 초점을 맞추어 검색할 수도 있을 것이고, 어떤 사람은 건물 자체에, 어떤 사람은 폭파되는 상황에 초점을 맞추는 사람이 있을 수 있을 것이다. 이러한 사용자의 다양한 요구는 질의 동영상 구성과 가중치 부여로서 만족될 수 있다. 사용자는 질의 동영상을 자유롭게 구성함으로써 자신이 원하는 문맥구조를 만들 수 있다. 그림 6은 저작 및 검색 도구에서 사용자가 질의 동영상을 구성한 것을 보여준다. 또한 전경, 배경, 오디오, 문맥 각각에 가중치를 부여함으로써 자신의 검색 의도가 무엇인지를 표현할 수 있으며 각각에 대해서 세부 가중치도 부여함으로써 좀 더 구체화시킬 수 있다.

4.2 실험 및 결과 분석

검색 시스템의 성능은 찾고자 하는 동영상을 얼마나 정확하고 빠짐없이 찾았는지의 정도로 표현할 수 있다. 이러한 측정 표준으로 많이 사용하는 것이 Precision과 Recall이다. Precision은 검색되어진 결과에 대해서 얼마나 정확하게 찾았는가를 나타내는 척도이고, Recall은 빠짐없이 찾았는가를 나타내는 척도이다. 그림 7은 Precision과 Recall을 이해하기 쉽게 그림으로 표현한 것이다.



$$\text{Precision} = \frac{B}{B+C}$$

$$\text{Recall} = \frac{B}{A+B}$$

A+B : 질의 샷과 관계 있는 샷
B+C : 검색도구에 의해서 검색된 샷

그림 7 성능 측정 척도

실험은 각기 다른 영화나 드라마에서 수집한 서로 다른 신 22개를 대상으로 수행하였다. 문맥 반영 여부를 나타내기 위해서 그 특징이 명확하지 않은 일반 신들과 폭발 신, 전쟁 신, 공포 신과 같이 저급 수준 내용 정보들의 패턴에 의해 특징이 명확해지는 신들을 균형 있게 사용하였다. 표 2는 실험에 사용한 22개의 신들에 대한 의미적인 특징을 설명한 것이다. 가중치는 전경, 배경, 오디오, 문맥에 대해서 부여할 수 있고, 각각의 세부 비교 요소들에 대해서도 가중치를 부여할 수 있다. 전경에 대한 세부 가중치는 Luminance, 움직임, 위치, 영역, 형태, 크기에 대해서 부여할 수 있으며, 배경에 대한 세부 가중치는 이미지 비교를 하기 때문에 사용되지 않는다. 오디오에 대한 세부 가중치는 진폭, 주파수, Zero Crossing에 대해서 각각 주어지며, 문맥에 대하여서는 Luminance, 진폭, 주파수, Zero Crossing, 움직임, 샷 길이에 대해서 주어지게 된다. 이렇듯 모든 비교 요소에 가중치를 부여하는 것은 사용자가 자신의 의도에 맞는 검색을 할 수 있도록 하기 위함이다.

실험은 가중치를 바꾸어 가면서 기대되는 결과와 실제 실험에 의한 결과를 비교하여 Precision과 Recall을

표 2 실험에 사용한 신

신 번호	신 특징	시간	샷 개수
S1	차량 폭발 신[엑스파일]	01'47"	20
S2	일반 신[건물탈출][매트릭스]	00'54"	17
S3	두명 대화 신[매트릭스]	00'39"	10
S4	대화 신[좋은 친구들]	00'31"	5
S5	대화 신[좋은 친구들]	00'28"	4
S6	일반 신[건물탈출][매트릭스]	00'51"	21
S7	일반 신 [텔미섬딩]	00'54"	23
S8	대화 신[텔미섬딩]	01'08"	8
S9	총 싸움 신[쉬리]	00'32"	10
S10	폭발 신[쉬리]	00'30"	11
S11	건물 폭발 신[쉬리]	00'25"	23
S12	일반 신[박하사탕]	01'12"	12
S13	일반 신[:박하사탕]	00'31"	3
S14	비행기 폭발 신[데스티네이션]	01'16"	49
S15	대화 신[엑스파일]	01'02"	10
S16	건물 폭발 신[엑스파일]	01'13"	32
S17	일반 신[시티오브엔젤]	00'35"	9
S18	대화 신[시티오브엔젤]	00'39"	13
S19	전쟁 신[왕건]	00'22"	16
S20	전쟁 신[왕건]	00'35"	15
S21	축구1	05'32"	47
S22	축구2	06'02"	45
합계	22 신	27'38"	403샷

구하는 방법으로 진행하였다. 예를 들어 S11은 영화 쉬리에 나오는 건물 폭발 신인데, 사용자가 건물이 폭발하는 장면을 중심으로 질의 신을 구성하고 사운드와 색상 비교에 높은 가중치를 부여하였다면, S1이나 S11, S16의 폭발 신에 포함된 건물이 폭발하는 샷이 검색되는 것이 바람직한 것이다. 이럴 경우 각각의 샷에 대한 비유사도가 일정 임계 값보다 낮게 되면 유사한 샷으로 결정하게 되고, 만일 유사한 샷에 세 개의 신에 포함된 건물 폭발 신이 모두 포함되면 Recall은 100이 되는 것이고, 단 3개만이 검색되었다면 Precision또한 100이 되는 것이다. 하지만 Precision과 Recall은 확실히 예상되는 결과 집합이 있어야만 측정 가능한 척도이기 때문에 폭발 신이나 대화 신과 같이 특별한 신들이 아니면 적용하기 어려울 수 있다. 따라서 <실험 1>에서는 폭발 신이나 대화 신과 같은 특징이 있는 신들에 대해서 검색 실험을 하였고 <실험 2>에서는 일반적인 신들에 대해서 검색 실험을 하였다.

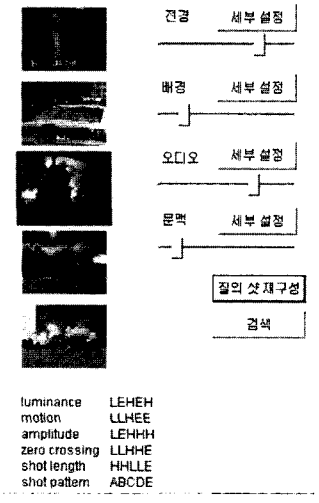


그림 8 <실험 1>질의 구성

• 실험 1

<실험 1>은 건물 폭발 신을 찾아내기 위해 질의 신을 구성하였다. 그림 8은 신 S16으로 구성한 질의 신 구성을 보여준다. 폭발 전의 정적을 표현하기 위해서 처음 두개의 샷을 배치했으며 이후 3개의 샷은 모두 폭발하고 있는 샷으로 배치하였고, 5개의 샷 중에서 가운데 샷과 유사한 샷을 결과 샷으로 검색하는 것이 목적이다. Luminance, 움직임, 진폭, Zero Crossing contour가 가운데 샷에서 갑자기 높아지는 것은 폭파가 시작되는 샷이라는 것을 보여주는 문맥이라고 할 수 있다. 폭발이 시작하는 샷을 찾기 위해서 가중치는 전경, 오디오, 문맥을 각각 4:3:1의 비율로 설정하였다. 세부적으로 보았

표 3 폭발이 시작되는 신과 샷

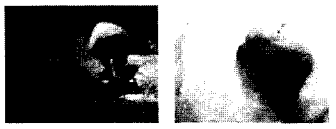
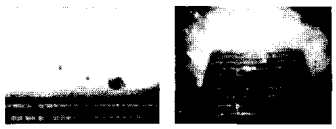
샷 번호	설명	샷의 키 프레임
S1신의 12 또는 13 샷	차량 폭발이 시작되는 샷	
S11신의 8 또는 9 샷	건물 폭발이 시작되는 샷	
S16신의 8 또는 9 샷	건물 폭발이 시작되는 샷	

표 4 <실험 1>검색 결과

순위	비유사도	샷 번호	설명	샷의 키 프레임
1	0.020803	S16 신 8 샷	건물 폭발이 시작되는 샷	
2	0.036337	S16 신 14 샷	건물이 폭파되는 샷	
3	0.036551	S1 신 12 샷	차량 폭발이 시작되는 샷	
4	0.037199	S6 신 10 샷	벌떼가 공격하는 샷	
5	0.037426	S16 신 17 샷	건물이 폭파되는 샷	
6	0.037823	S6 신 6 샷	벌떼의 공격이 시작되는 샷	

을 때, 폭발 신의 특징은 Luminance와 오디오 진폭의 큰 증가와 영역 색상이 비슷하고 샷 전체의 움직임이 증가하는 반면에, 객체 형태 정보나 배경 정보는 의미가 없어지게 된다. 검색 시스템의 성능 측정 척도인 Precision과 Recall을 구하려면 예상되는 검색 결과가 있어야 한다. 표 3은 예상 결과인 폭발이 시작되는 샷을 보여준다. 이외에도 비행기 폭파와 사람 몸을 폭파하는 샷이 있지만 폭파 효과가 질의 동영상과 확연히 차이가

나므로 예상 결과에서 제외하였다.

실험 결과 비유사도 임계 값을 0.04로 했을 때 총 6개의 샷이 검색되었으며 표 4는 실제 검색된 샷을 보여준다. 따라서 <실험 1>의 Precision은 0.33이며, Recall은 0.66의 성능을 갖는다. 검색 결과를 살펴보면 만일 임계 값을 0.037로 설정을 하였다면 Precision 값 역시 0.66이 되는 것을 알 수 있다. 검색결과는 비유사도가 낮은 것부터 순서대로 나열하기 때문에 Precision보다는 Recall 값이 더 의미가 있음을 알 수 있다. 또한 가중치 설정에서 문맥에 대한 가중치를 0으로 조정하면, 검색 결과에서 S6의 6, 10 샷이 사라지고 불길이 치솟는 샷만을 검색해 낸다. S6의 6, 10샷은 움직임과 Luminance, 진폭이 큰 폭으로 변하여 검색된 샷들이기 때문에 문맥 정보를 반영하지 않고 전경만을 반영하게 되면 질의 샷과의 비유사도가 많이 올라가기 때문이다. 실험 결과에서 볼 수 있는 바와 같이 가중치를 어떻게 부여하느냐에 따라서 다양한 결과를 낼 수 있으며, 두개의 샷이 비슷하다는 것은 보는 관점에 따라 틀려질 수 있기 때문에 검색된 샷이 옳거나 그르다고 단정할 수 없다는 문제가 있다. 하지만 본 실험은 일반적인 상식선에서 예상되는 결과 샷을 선정했으며, 선정된 샷을 바탕으로 Recall과 Precision을 구하였다. 구현된 시스템을 사용하여 대화 장면, 폭발장면과 같이 가지고 있는 특징이 뚜렷하여 실험 결과를 예상할 수 있는 것들에 대해서 3개의 질의 동영상을 더 만들어 실험을 한 결과 Precision 값은 0.3대로 매우 낮은 값을 보여주지만 Recall은 0.6에서 0.8까지 상당히 높은 검출 율을 보여주었다.

• 실험 2

<실험 2>는 폭발 신이나 대화 신과 같이 저급 수준 내용 정보들이 특별한 패턴을 가지고 있지 않아서 예상 결과를 확실하게 알 수 없는 질의 동영상을 구성하여 실험하였다. 따라서 <실험 2>는 Precision이나 Recall에 대한 분석보다는 결과 동영상이 검색된 이유를 중심으로 분석하였다. 그림 9는 질의 동영상의 구성을 보여주는 그림이다. 첫 번째 샷은 축구 경기 중 줌 아웃(Zoom Out) 샷을 배치하였고 두 번째 샷은 같은 경기의 줌 인(Zoom In) 샷을 배치하였다. 가중치는 전경, 배경, 문맥을 고르게 설정하였으며, 표 5는 이와 같은 조건 하에서의 검색 결과를 12개까지 보여준다.

결과 샷을 분석해보면, 우선 모든 샷이 줌 인 샷임을 알 수 있다. 이것은 전경의 형태와 움직임 비교에 의해 검색되어진 것이다. 일반적으로 운동 경기의 줌 인 샷은 움직임과 형태가 매우 큰 특징이 있다. 하지만 질의 샷과 같은 줌 인 샷은 축구 경기 뿐 아니라 다른 모든 동영상에서도 쉽게 찾을 수 있는 장면임에도 결과 샷은 모두 축구 경기 신에 잇는 샷이다. 이는 녹화나 녹음 상

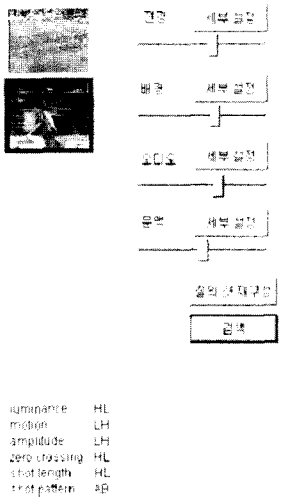


그림 9 질의 구성

태에 따라서 큰 영향을 받는 Luminance와 오디오의 영향을 받은 것으로 생각할 수 있다. 또한 축구 경기의 오디오는 지속적으로 아나운서의 음성과 관객의 함성 소리가 함께 나온다는 점에서 다른 동영상과 구별되어 지며 축구 경기 중에 나오는 그라운드나 관람석과 같은 배경도 결과 샷이 축구 경기 신에서만 나온 이유로 설명할 수 있다. 줌 인 샷과 줌 아웃 샷은 움직임 크기와 분포 면에서 큰 차이가 난다. 결과 샷 12개중 줌 아웃 샷 뒤에 나오는 줌 인 샷은 모두 9개로서 문맥 중 움직임에 관한 부분이 반영된 것으로 생각할 수 있다.

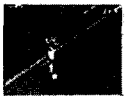
4.3 기존 연구와의 비교

기존의 다른 예제 기반 동영상 검색 시스템들과 비교해보았을 때 본 시스템이 가지는 가장 큰 장점은 문맥 정보를 검색에 반영하였다는 것과 사용자의 검색 의도를 쉽게 반영할 수 있도록 비교요소를 정의했다는 것이다. 문맥을 반영함으로써 의미정보를 검색에 적용할 수 있게 되었고, 비교 요소를 전경, 배경, 오디오, 문맥으로 나누어 사용자가 가중치를 부여할 수 있게 하고, 각각에 대해서 세부 가중치를 두어 미세한 부분까지 조정할 수 있게 하여서 사용자 의도에 맞는 검색을 할 수 있도록 하였다. 기존의 예제기반 동영상 검색 방법을 살펴보면 다음과 같다.

4.3.1 대표 프레임 비교 방법

M.M.Yeung의 연구[5]에서는 대표 프레임 단위의 비교 방법을 사용한다. 대표 프레임 비교 방법에서는 동영상에서 대표되는 몇 개 프레임의 정보만을 가지고 비교를 수행하기 때문에 비교 과정에서 시간 정보는 손실되게 된다. 하지만 일반적으로 동일한 샷 안의 프레임들은 큰 변화를 갖지 않기 때문에 동영상이 샷 단위로 이루

표 5 <실험 2> 검색 결과

순위	샷 번호	키 프레임	순위	샷 번호	키 프레임
1	S21신 10샷		2	S21신 41샷	
3	S21신 14샷		4	S21신 46샷	
5	S22신 18샷		6	S22신 4샷	
7	S22신 10샷		8	S22신 12샷	
9	S22신 15샷		10	S22신 1샷	
11	S22신 13샷		12	S22신 6샷	

어져 있다면 이러한 비교 방법이 사용될 수 있는 근거를 갖는다. 비교 방법은 다음과 같다. 두개의 동영상 샷이 있을 때 그 사이에 유사한 프레임 쌍이 존재할 때 두개의 샷을 유사하다고 하며, 유사함의 정도는 비 유사도 값으로 표현하는데 두개의 샷에서 가능한 모든 대표 프레임들의 거리(Euclidean distance)를 조사한 후 가장 작은 값을 두 샷의 비 유사도 값으로 정의하게 된다. 이 방법은 QBE 시스템에 있어서 가장 간단한 방법으로서 현재에는 거의 사용되지 않는 방법이다.

4.3.2 객체 비교 방법

H.Zhang의 연구[6]에서 비교 연산은 객체 단위로 이루어진다. 색상과 움직임의 일관성을 사용하여 추출된 객체에 대하여 객체 관련 정보들을 추출한다. 예를 들면, 형태(shape), 재질(texture), 색상(Color), 움직임(Motion) 등을 추출하여 사용하는데, H.Zhang은 이들을 객체 구조 체로 구성하고 유사도 비교 대상으로 사용하였다.

제안하고 있는 비교 방법은 다음과 같다. 먼저 각각의 샷에서 객체를 추출하고 전술한 객체 구조 체로서 표현을 한 후, 다수의 객체가 있을 경우 객체들간에 일대일 비교를 하여 그 유사도가 가장 높은 것이 임계 값 이상이면 비슷한 샷으로 선택한다.

4.3.3 VideoQ 시스템

VideoQ 시스템[7]은 객체의 모양과 움직임을 중심으로 하여 비교하는 방법이다. 두개의 동영상에 대해 전체 비유사도(Distance)는 다음에서 정의한 각 비교요소의 비 유사도 값에 가중치를 적용한 합으로서 정의한다.

• 객체의 움직임 비교

움직임에 대한 비교는 두 가지 형태의 비교 작업이 가능하다. 첫 번째는 공간적인 정보만을 사용하여 비교하는 방법이고, 두 번째는 공간-시간 정보 비교 방법이다. 전자는 객체의 움직임 시간을 정확히 알지 못할 때 사용할 수 있는 방법이고, 후자는 시간대 비교까지 수행하기 때문에 조금 더 정확한 비교가 가능하다. 비교대상의 길이가 틀릴 경우에는 짧은 동영상에 맞추거나 정규화 해서 비교 작업을 한다.

• 객체의 색상 비교

색상은 CIE-LUV 영역으로 바꾸어 Euclidean Distance를 구하여 사용한다.

• 객체의 재질 비교

굵기, 명암, 방위 3가지 재질 요소의 가중치를 갖는 Euclidean Distance를 사용하여 두 객체간의 재질을 비교한다.

• 객체의 형태 비교

객체의 Aspect Ratio를 비교한다.

• 객체의 크기 비교

전체 화면에 대해 객체가 차지하고 있는 영역의 백분율을 구하여 비교한다.

표 6 검색 방법 비교

	비교 단위	비교 방법	사용자 의견 반영
동적 프로그래밍[9]	색상, 오디오, 움직임, 외곽선	모든 프레임을 비교하며 동적 프로그래밍 방법을 사용	가중치 부여 방법
다항식 점근법[8]	색상	일정 윈도우 단위로 비교하며, 윈도우 내부는 다항식 점근법을 사용하여 유추.	-
VideoQ[7]	객체	객체의 특징과 움직임을 중심으로 비교	가중치 부여 방법
대표 프레임 비교 방법[5]	대표 프레임	샷에서 한 개 또는 다수의 대표 프레임을 선택하여 이미지비교	-
객체 비교 방법[6]	객체	객체의 특징을 구조 체로 구성하여 비교	-
문맥 고려 방법	전경, 배경, 오디오, 문맥	샷을 분석하여 대표 전경과 대표 배경을 분리하여 비교하고 인접한 샷들로 만들어진 contour로 문맥 비교	가중치 부여 방법

4.3.4 다항식 점근법 비교 방법

M.R.Naphade의 연구[8]에서는 사용하는 저급 수준 정보는 색 히스토그램 정보 하나만을 사용하고 있으나 새로운 비교 방법을 제안하고 있다는 면에서 의미를 찾을 수 있는 연구이다. 동영상 비교를 하는데 있어서 가장 좋은 방법은 대상 동영상과 질의 동영상의 모든 프레임을 일대 일로 비교하는 방법일 것이다. 하지만 이러한 방법을 사용하면 계산 량이 매우 커지기 때문에 실제적으로 사용하기에는 현실적이지 못하게 된다. 이러한 문제를 해결하기 위해서 M.R.Naphade는 다항식 점근법이라는 알고리즘의 사용을 제안하고 있다. 다항식 점근법은 이산 적인 값들의 나열이 있을 때 그 사이 값들을 특수한 다항식을 통해서 추측해 구하는 방법이다. 실제로 비교하는 프레임들의 사이 값들을 다항식 점근법을 통해서 구함으로써 계산 속도를 높임과 동시에 모든 값들을 계산했을 때와 거의 비슷한 결과를 구해낼 수 있게 된다.

4.3.5 동적 프로그래밍 비교 방법

M.R.Naphade 의 또 다른 연구[9]에서는 동적 프로그래밍 방법을 사용하고 있다. 이 연구에서는 동적 프로그래밍 방법을 대상 동영상의 모든 프레임의 가능한 조합을 비교하려고 했을 때 비교 시간이 너무 많이 걸리는 단점을 해소 할 수 있는 알고리즘으로서 제시하고 있다. 동적 프로그래밍 방법을 사용해서 색 히스토그램, 움직임 히스토그램, 외곽선 방향 히스토그램, 그리고 오디오 정보, 총 4가지 비교 요소에 대해서 비유사도를 구해낸다.

표 6은 기존의 방법과 본 논문에서 제시한 비교 방법에 대한 차이를 정리한 것이다.

5. 결론 및 앞으로의 연구 방향

디지털 동영상 데이타를 효율적으로 관리하기 위해서 디지털 비디오 라이브러리(Digital Video Library : DVL)가 구축되고 있으며 이러한 DVL 시스템에 반드시 필요한 기능중의 하나로 예제 기반 동영상 검색을 들 수 있다. 본 논문에서는 이웃 한 샷간의 패턴변화를 고려한 새로운 예제 기반 동영상 알고리즘을 제안하였고 이를 바탕으로 하여 동영상 검색 시스템을 구현하였다.

본 논문에서는 기존의 예제 기반 동영상 검색 시스템들이 샷 단위로 객체나 프레임 전체를 중심으로 비교하는 것에 그쳤던 것과는 달리 문맥 즉, 이웃 한 샷간의 관계를 비교요소로서 사용함으로써 다양한 사용자의 검색 요구를 반영할 수 있도록 하였다. 이러한 시도는 양적이 비교만을 했던 기존의 예제 기반 동영상 검색 시스템과는 달리, 문맥을 반영 함으로서 기초적이지만 의미적인 요소를 동영상 검색에 적용했다는 면에서 의미를 찾을 수 있을 것이다. 스토리를 가진 동영상에서 신은 이웃하고 의미적으로 연관을 가지고 있는 샷들의 모임으로 생각할 수 있다. 따라서 하나의 신을 구성하는 샷 간의 저급 수준 내용 정보 사이에는 어떤 패턴을 가질 수 있으며 이는 샷의 중의성을 해결할 수 있는 중요한 단서로서 사용될 수 있다. 샷 간의 패턴이 의미를 갖는 저급 수준 내용 정보로는 Luminance, 움직임, 진폭, 주파수, 샷 길이, 샷 유사도 가 있으며 이들은 contour형식으로 표현되어 비교요소로서 사용되었다. 또한, 기존의 예제 기반 동영상 검색에서는 거의 사용하지 않았던 오디오 정보를 비교요소로 사용함으로써 어두운 장면이나 오디오가 강조되는 샷을 찾는데 보완 할 수 있었다. 또한 동영상의 제작 방법과 같이 객체 또는 프레임 단위의 비교가 아닌 전경과 배경에 관련된 것으로 나누어 비교를 수행하였다. 사용자가 질의 신을 구성하고, 각각의 비교요소에 가중치를 부여함으로써 자신의 검색 의도를 최대한 자유롭게 반영할 수 있게 하였다. 검색 시스템에 있어서 데이타가 많아짐에 따라 비교 시간 또한 매우 큰 문제가 될 수 있다. 더욱이 동영상 데이타는 정보량이 매우 크기 때문에 모든 프레임을 하나씩 비교하는 것은 매우 많은 시간을 필요로 하게 된다. 따라서 샷을 대표할 수 있

는 객체를 찾기 위해 샷 안에서 가장 오랫동안 유지되는 영역을 모아 대표 전경을 구성하였으며 이를 뺀 나머지를 대표 배경으로 하여 저급 수준 내용 정보를 구성하였다. 물론 이러한 방법은 샷 안에서 카메라 이동이나 객체의 큰 움직임 등에 대해서는 많은 정보를 손실할 수 있지만 대표 정보만을 저장하고 큰 변화가 있는 정보는 따로 저장함으로써 이러한 문제를 최소화하면서 비교 시간을 절약할 수 있었다.

본 논문에서 앞으로 해결해야 할 부분은 크게 2가지로 살펴볼 수 있다. 첫 번째로 오디오 정보를 비중 있게 이용할 수 있는 방안을 연구해야 한다. 오디오는 비디오와 함께 동영상을 구성하고 있는 매우 중요한 요소이며 이에 대한 연구는 검색 성능을 높이기 위해서 반드시 필요하다. 두 번째로 개선해야 할 부분은 검색 속도에 관한 부분이다. 좋은 동영상 검색 시스템이 되기 위해서는 사용자가 원하는 동영상을 최대한 정확하고 빠르게 찾아 주어야 할 것이다. 이를 해결하기 위해서 효율적인 DB 인덱싱 방법이나 각 비교요소들에 특화된 빠른 비교 방법에 관한 연구가 필요하다. 이러한 두 가지 개선사항 외에도 개선해야 할 많은 점이 있으며, 이러한 부분들을 하나씩 개선해 나간다면 보다 나은 예제 기반 동영상 검색 시스템을 구현할 수 있을 것이다.

참 고 문 헌

[1] M. Flickner, et.al., "Query By Image and Video Content: The QBIC System," *IEEE Computer Magazine*, Vo.28, pp.23-32, 1995.

[2] ISO/IEC JTC1/SC29/WG11 *Information Technology-Multimedia Content Description Interface-Part3: Visual*, 2001.

[3] ISO/IEC JTC1/SC29/WG11 *Information Technology-Multimedia Content Description Interface-Part4: Audio*, 2001.

[4] D.Zhong and S.F.Chang, "An Integrated Approach for Content Based Video Object Segmentation and Retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.9, No.8, pp.1259-1268, 1999.

[5] M.M. Yeung and B.Liu, "Efficient Matching and Clustering of Video Shots," *Proceeding of the IEEE International Conference on Image Processing*, Vol.1, pp.338-341, 1995.

[6] H.Zhang, A.Wang and Y.Altunbask, "Content Based Video Retrieval and Compression : A Unified Solution," *Proceeding of the IEEE International Conference on Image Processing*, Vol.1, pp.13-16, 1997.

[7] S.F Chang, W. Chen, H.J.Meng, H.Sundaram and D.Zhong, "A Fully Automated Content Based Video Search Engine Supporting Spatio-Temporal Queries," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.8, No.5, pp.602-615, 1998.

[8] M.R.Naphade, M.M.Yeung and B.L.Yeo, "A Novel Scheme for Fast and Efficient Video Sequence Matching using Compact Signature," *Proceeding of SPIE Storage and Retrieval for Media Database 2000*, Vol.3972, pp.564-572, 2000.

[9] M.R.Naphade, R. Wang and T.S. Huang, "Audio-Visual Query and Retrieval : A System that Uses Dynamic Programming and Relevance Feedback," *Jounal of Electronic Imaging*, Vol.10, pp.861-870, 2001.

[10] N.V.Patel and I.K.Sethi, "Video Shot Detection and Characterization for Video Databases," *Proceeding of SPIE Storage and Retrieval for Image and Video Database*, pp.218-225,1997.

[11] F.Long, et al., "Extracting Semantic Video Objects," *IEEE Computer Graphics and Applications*, Vol.21, pp.48-55, 2001.

[12] R.J.Qian, P.J.L. van Beek and M.I.Sezan, "Image Retrieval Blob Histogram," *Proceeding of IEEE Multimedia and Expo 2000*, Vol.1, pp.125-128, 2000.

[13] S.Pfeiffer, S.Fischer and W.Effelsberg, "Automatic Audio Content Analysis," *Proceeding of ACM Multimedia 96*, pp.21-30, 1996.

[14] S.D.Katz, *Film Directing : Shot by Shot*, Michael Wiese Production, 1991.

[15] 김재홍, *MPEG 시스템 스트림 상에서 오디오 정보를 이용한 신경계 검출 방법*, 서강대학교 석사 학위 논문, 1998.

[16] S.Blackburn and D.Deroure, "A Tool for Content Based Navigation of Music," *Proceeding of ACM Multimedia 98*, pp.361-368, 1998.

박 주 현

1999년 서강대학교 전자계산학과(공학학사). 2002년 서강대학교 컴퓨터학과(공학석사). 2002년~서강대학교 컴퓨터학과 (박사과정)

낭 종 호

1986년 서강대 전자계산학과 졸업. 1988년 한국과학기술원 석사. 1992년 한국과학기술원 박사. 1992년~1993년 Fujitsu 연구소 연구원. 1993년~현재 서강대학교 컴퓨터학과 교수

김 경 수

1983년 서울대학교 공과대학 제어계측공학과 졸업. 1985년 서울대학원 제어계측공학과 석사. 1985년~현재 KBS기술연구소 연구원 관심분야는 멀티미디어 방송, 제작시스템, 비디오 인덱싱, 영상처리

하 명 환

1993년 경북대학교 공과대학 전자공학과 졸업. 1995년 한국과학기술원 전기전자공학과 석사. 1995년~현재 KBS 기술연구소 연구원. 관심분야는 멀티미디어 방송, 제작시스템, 비디오 인덱싱, 영상처리

정 병 희

1994년 이화여자대학교 전자계산학과 졸업. 1996년 한국과학기술원 전산학과 석사. 1996년~현재 KBS 기술연구소 연구원. 2000년~현재 한국과학기술원 전산학과 박사과정 재학중. 관심분야는 멀티미디어 방송시스템, 미디어 아카이브, 초고속네트워크 시스템