

요약 비디오 영상과 PCA를 이용한 유사비디오 검출 기법

유재만[†], 김우생^{**}

요 약

웹상의 출판이 보편화 될수록 많은 데이터의 내용물들이 압축, 포맷, 편집 등 변형된 상태로 중복해서 존재하게 된다. 이러한 유사한 데이터들은 검색 시 속도나 검색률 등에 문제를 야기 시킬 수도 있으며, 반면에 특정 사이트에 문제가 발생할 경우 다른 사이트의 중복된 데이터를 제공해 줄 수도 있게 된다. 따라서 본 논문에서는 대규모 데이터베이스 상에 존재하는 비디오들 중에서 유사한 데이터들에 대한 정보를 사전에 감지할 수 있는 효율적인 방법을 제안한다. 본 연구에서는 비디오들을 직접 비교하는 대신 비디오를 대표하는 요약 비디오 영상을 만들고, 주성분 분석(PCA-principle component analysis) 기법을 적용하여 저차원 특징 벡터 상에 군집화를 통해 유사 비디오들을 검출하였다. 실험을 통하여 제안하는 방법의 효율성과 정확성이 우수함을 보였다.

Similar Video Detection Method with Summarized Video Image and PCA

Jaeman Yoo[†], Woosaeng Kim^{**}

ABSTRACT

With ever more popularity of video web-publishing, popular content is being compressed, reformatted and modified, resulting in excessive content duplication. Such overlapped data can cause problem of search speed and rate of searching. However, duplicated data on other site can provide alternatives while specific site cause problem. This paper proposes the efficient method, for retrieving similar video data in large database. In this research we have used the method to compare summarized video image instead of the raw video data, and detected similar videos through clustering in that dimension feature vector through PCA(principle component analysis). We show that our proposed method is efficient and accurate through our experiment.

Key words: Video(비디오), Video Retrieval(비디오 검색), Similar Video Detection(유사비디오 검색)

1. 서 론

인터넷의 발달과 함께 웹상의 정보가 기하급수적

으로 증가하고 있으며 이와 동시에 많은 중복된 데이터가 웹상에 필연적으로 존재하게 된다. 웹에 있는 모든 텍스트 문서의 46%는 포맷 등 낮은 단계의 내용만 조금씩 다른 적어도 하나의 유사한 문서를 갖고 있으며, 전체 문서의 5%는 10개에서 100개의 중복된 문서들을 갖고 있다고 조사되었다[1]. 특히 동영상의 경우에는 저작물 틀들의 발달과 함께 사용자가 웹상의 정보를 검색할 뿐 아니라 수정, 편집해 추가하는 경우도 점점 많아지고 있다. 따라서 웹상에는 같은

※ 교신저자(Corresponding Author): 김우생, 주소: 서울특별시 노원구 월계동 447-1번지(139-701), 전화: 02)940-5217, FAX: 02)909-0998, E-mail: kwsrain@cs.kw.ac.kr, 접수일: 2004년 6월 2일, 완료일: 2004년 8월 11일

[†] 광운대학교 컴퓨터 공학부

(E-mail: semyoo3@cs.kw.ac.kr)

^{**} 정회원, 광운대학교 컴퓨터 공학부

※ 이 논문은 2003년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

비디오들이라도 내용의 일부가 편집되거나, 압축 방식이 다르거나, 크기나 프레임 비율이 다르게 포맷된 상태의 유사한 데이터들이 많이 중복되어 존재하게 된다.

이와 같은 환경에서 유사한 비디오들의 정보를 아는 것은 다음과 같은 여러 가지 이유에서 필요하다. 우선 멀티미디어 검색은 중복된 유사 비디오들도 함께 검색하게 되므로 속도나 검색 정확도 등에서 문제가 야기되며, 따라서 이러한 중복된 데이터들을 사전에 감지할 수 있는 방법이 필요하다. 이러한 중복된 데이터들의 정보를 알고 있으면 사용자의 요청에 맞게 위치 등을 고려한 가장 적합한 비디오를 제공할 수 있을뿐더러, 만약 특정 사이트나 회선에 문제가 발생한 경우 다른 사이트에 있는 유사한 비디오 정보도 제공해 줄 수도 있다. 또한 멀티미디어 제작측면에서는 저작권 상의 문제나 사업상등의 문제에 근거하여 웹상에 같거나 유사한 데이터들이 이미 존재하고 있는지도 관심의 대상이 될 것이다.

본 논문에서는 웹 등의 대규모 데이터베이스 상에 존재하는 비디오들 중에서 유사한 비디오들을 효율적으로 추출하는 방법을 제안한다. 대규모 데이터베이스에 적용하기 위해서는 무엇보다도 낮은 복잡도를 갖는 것이 필요하다. 따라서 본 연구에서는 비디오들을 직접 비교하는 대신 비디오의 내용을 대표하는 하나의 요약 비디오 영상을 구하여 이들의 유사성을 비교하는 새로운 방법을 제안한다. 요약 비디오 영상은 비디오 전체에서 일부 샘플을 취해 구성된 하나의 영상이기 때문에 비디오들을 직접 비교하는 것에 비해 수행속도가 현저히 줄어든다. 또한 본 연구에서는 요약 비디오 영상들의 특징 벡터공간의 차수를 줄이기 위하여 주성분 분석 기법(PCA-principle component analysis)을 적용하였으며, 이를 통한 저차원 특징벡터 공간상에 군집화를 통하여 유사 비디오들을 효율적으로 찾아내는 방법을 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 연구배경에 대해서 설명하고, 3장에서는 요약 비디오 영상을 생성하는 방법과 주성분 분석과 군집화를 사용하여 저차원 특징벡터 공간상에서 요약 비디오 영상들을 구분하는 방법에 대하여 설명하며, 4장에서는 구현 및 실험 결과를 보여주고, 마지막으로 5장에서는 결론 및 추가 연구에 대하여 언급하였다.

2. 연구 배경

데이터베이스에 있는 비디오들 중 질의 비디오 클립과 매칭 되는 비디오를 찾는 방법에서는 비디오 클립의 연속된 프레임들을 fingerprint라는 연속된 특징벡터로 부호화한 후 그들 사이의 유클리드 거리를 계산함으로써 매칭을 시도하였다[2]. 비디오 데이터로부터 fingerprint를 계산해냄으로써 비디오 데이터들의 실시간 매칭을 빠르게 할 수 있으나, 하나의 비디오를 대표하는 fingerprint를 구성하기 위해 전체 프레임을 사용해야 한다는 단점이 있다. 압축된 비디오의 DCT에 있는 영상의 히스토그램들을 추출하여 이들의 변형된 교집합을 통해 비디오들의 매칭을 시도하는 방법에서는 연속된 프레임들의 히스토그램들은 비슷하다는 성질을 이용하여 다항식 근사법을 적용하여 계산량을 줄였다[3]. 이 방법 역시 하나의 비디오를 대표하기 위해 여러 개의 프레임을 사용하게 된다는 단점이 있다.

비디오 프레임의 영상 정보에 기반한 시그니처(signature) 비교에 의해 비디오의 유사성을 검사하는 방법에는 두 가지가 있다. 첫 번째로는 비디오 시퀀스간의 매칭을 패턴 매칭 문제로 바꾸어서, 특징값들로 표현되는 비디오 시퀀스를 심볼의 시퀀스로 변환시켜 유사성 매칭을 시도하는 방법이 있고[4], 두 번째로는 여러 개의 시드(seed) 이미지와 각 비디오의 프레임을 컬러 히스토그램 정보를 이용하여 비교하는 방법으로 유사 비디오의 동일 프레임을 각 비디오의 시그니처 프레임으로 추출하여 비디오의 유사성을 시그니처들 간의 비교로 측정하는 방법이 있다[5]. 위의 두 가지 방법은 비디오의 모든 프레임의 영상정보를 이용하는 비교 방법에 비해 비교할 데이터의 양을 감소시키는 장점이 있으나, 반면에 시그니처 프레임의 결정이 시드 이미지의 특정 영상 정보에 민감하게 반응한다는 단점이 있다. 예를 들어 시드 이미지가 바다의 영상 정보에 민감하다고 하면 이는 시그니처 생성시 서로 다른 비디오의 바다 장면 프레임을 그 비디오의 시그니처로 결정하게 되고, 컬러 히스토그램에 의해 동일한 비디오로 결정할 수 있다. 또한 위의 비디오 시그니처를 사용하는 방법에 각각의 비디오 클립들과 연관된 텍스트 정보나 하이퍼링크 정보인 메타 데이터를 이용하는 방법을 더하

여 유사비디오를 검색함으로써 성능을 높일 수 있을 것으로 보였으며[6], 비슷한 시그니처들을 추상적인 임계 그래프(abstract threshold graph)에서 완전히 연결된 요소(complete connected component)들로 군집화하는 방법을 통하여 유사한 비디오들을 구분하는 알고리즘이 제안되었다[7]. 하지만 이러한 모든 방법들은 하나의 비디오를 대표하기 위해 여러 개의 프레임들 또는 그들의 특징벡터들을 사용해야 한다는 문제점이 있다. 또한 시그니처를 사용하는 방법은 비디오의 유사성 검사를 위해 시그니처를 추출하여 사용함으로써 사용할 데이터의 양을 줄인다는 장점이 있지만 정확한 유사성 판별에 문제점이 발생할 수도 있는 단점이 있다.

본 논문에서는 비디오의 영상정보를 요약 추출하여 이를 분석함으로써 유사 비디오에 대한 동일성 판별 방법을 제시한다.

3. 요약 비디오 영상과 유사 비디오 검출 방법

유사 비디오들을 찾아내는 전체적인 과정이 그림 1에 설명되어 있다. 시스템의 구성은 크게 요약 비디오 영상 생성 단계, 특징벡터 추출단계, 주성분 분석 단계, 클러스터링 단계로 구분된다. 요약 비디오 영상 생성 단계에서는 비디오의 각 프레임으로부터 시드 이미지와 가장 유사한 프레임을 찾아내고, 추출된 유사 프레임의 대각 픽셀을 모아 요약 비디오 영상을 구한다. 특징벡터 추출단계에서는 각 요약 비디오 영상으로부터 특징 벡터를 구한 후, 이들을 세로 방향으로 차례대로 나열하여 하나의 행렬을 구성한다. 따라서 행렬의 세로 방향은 특징벡터의 크기가 되고, 가로 방향은 전체 비디오들의 숫자가 된다. 다음으로 주성분 분석 단계에서는 각 세로 방향의 특징 벡터들에 주성분 분석을 적용하여 구한 저차원 벡터 공간상에 특징 벡터들을 투영한다. 마지막으로 클러스터링 단계에서는 투영된 특징 벡터들의 군집화를 통하여 유사 비디오들을 구분해 낸다. 각 과정에 대한 자세한 설명은 다음 장에서 하고 있다.

한 설명은 다음 장에서 하고 있다.

3.1 요약 비디오 영상 생성

요약 비디오 영상은 비디오를 대표하는 하나의 영상이다. 요약 비디오 영상은 우선 주어진 비디오에서 일부 프레임들을 선정한 후, 각 프레임의 특정 픽셀들을 추출하여 이들을 공간적으로 나열하여 생성한다. 만약 비디오에서 M_1 프레임을 선정하고 각 프레임에서 M_2 픽셀들을 추출한다면 요약 비디오 영상의 크기는 $M_1 * M_2$ 가 된다. 따라서 요약 비디오 영상은 3차원 데이터의 2차원 추상화라고 볼 수 있다. 크기(전체 프레임수)가 W_1, W_2 인 두개의 비디오의 유사성을 조사하는 단순한 방법은 모든 프레임들의 유사성을 서로 조사하는 것으로 $O(W_1 * W_2)$ 의 비교가 필요하다. 반면 제안하는 방법에서는 한 쌍의 요약비디오 영상들만을 조사하면 됨으로 $O(1)$ 의 비교로 줄어 들게 된다.

어떤 프레임과 어떤 픽셀들을 추출하느냐에 따라 여러 가지 요약 비디오 영상이 생성될 수 있다. 그러나 요약 비디오 영상은 다음의 두 가지 조건을 만족해야 한다. 유사 비디오들로부터 생성된 요약 비디오 영상들은 서로 유사해야 하며, 서로 다른 비디오들로부터 생성된 요약 비디오 영상들을 서로 달라야 한다. 이러한 조건들을 만족하며 동시에 비디오의 압축, 포맷 등 낮은 단계의 변형에도 강건한 요약 비디오 영상들을 구하기 위하여 우리는 다음과 같은 방식을 사용한다. 먼저 랜덤하게 선택된 하나의 시드 영상과 유사한 M 개의 비디오 프레임들을 선택한다. 시드 영상과 비디오 프레임 영상들을 $X * Y$ 픽셀의 칼라 이미지라고 하고 $f^p(x,y)$ 를 칼라 채널 $P \in RGB$ 상의 영상 f 의 픽셀 (x,y) 에 있어서의 밝기라 할 때, 시드 영상 S 와 유사한 프레임 F 의 유사도 $U(S,F)$ 는 (1)과 같다.

$$U(S,F) = \frac{1}{X \cdot Y} \sum_{P \in \{R,G,B\}} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} |S^P(x,y) - F^P(x,y)| \quad (1)$$

비디오의 경우 비슷한 내용의 프레임들이 연속하여 모여 있기 때문에, 시드 영상 S 와 가장 유사한 영상들이 한곳에 몰려 있을 가능성이 많다. 따라서 프레임 비율이 서로 다르게 포맷된 유사 비디오들의 경우에 시드를 통하여 선택하게 되면 추출된 영상들의 일부가 서로 다를 가능성이 발생할 수 있다. 이러

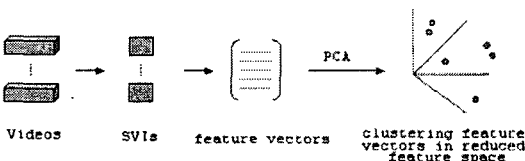


그림 1. 유사 비디오를 조사하는 전체적인 과정

한 경우를 방지하기 위해 본 연구에서는 하나의 비디오에서 시드 영상 S 와 유사한 M 개의 프레임을 선택할 경우 시스템은 우선 비디오 V 를 $V=V_1UV_2\cdots UV_L$ 의 L 개의 (단 L 은 M 의 약수이다. 각 블록에서 같은 수의 프레임을 추출한다.) 블록으로 나눈 후 각 블록에서 (2)를 만족하는 M/L 개의 프레임들을 추출한다.

$$F_i = \arg \min_{F \in V_i} d(S, F) \quad (2)$$

선택된 각 프레임에서 어떤 특정 그룹의 픽셀들을 샘플링 하는가에 따라 여러 가지 종류의 요약 비디오 영상이 생성될 수 있다. 예를 들어 각 프레임의 중간에 위치한 가로 방향이나 세로 방향 또는 대각선 방향의 픽셀들을 샘플링 할 수 있다. 그러나 본 연구에서는 각 프레임의 대각선 방향의 픽셀들이 원래의 비디오 데이터의 비주얼한 특징들을 가장 잘 나타낸다는 연구 결과에 따라 대각선 방향의 픽셀들을 샘플링 하였다[8]. 따라서 요약 비디오 영상의 세로방향의 픽셀들은 선택된 각 프레임의 대각선 방향 픽셀들로 구성된다.

그림 2는 3개의 서로 다른 비디오 클립으로부터 생성된 요약 비디오 영상들을 보여준다. 그림 2 (a)와 (b)는 유사한 비디오 클립으로부터 생성된 요약 비디오 영상들이며, 반면에 그림 2 (c)는 다른 비디오 클립으로부터 생성된 요약 비디오 영상이다. 유사 비디오들로부터 생성된 요약 비디오 영상들의 모양은 유사하며 다른 비디오로부터 생성된 요약 비디오 영상의 모양은 다른 것을 알 수 있다.

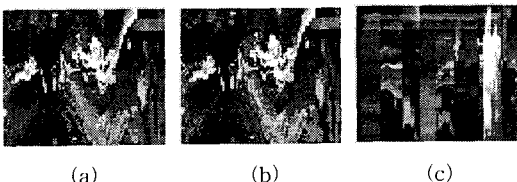


그림 2. 서로 다른 비디오 클립으로부터 생성된 요약 비디오 영상

3.2 주성분 분석을 통한 저차원 특징 벡터 공간

요약 비디오 영상들을 비교하기 위하여 영상에서 특징벡터를 추출한다. 영상의 특징을 추출하는 간단한 방법은 컬러 히스토그램이지만, 컬러 히스토그램은 영상의 공간적인 정보를 표현할 수 없기 때문에,

본 연구에서는 영상의 컬러 분포뿐 아니라 공간정보도 표현할 수 있는 영역 분할된 컬러 히스토그램 기법을 사용하였다[8]. 이를 위하여 요약 비디오 영상을 4개의 블록으로 나누고 각 블록에서 R, G, B의 최상의 2비트를 통하여 구해진 64빈 컬러 히스토그램을 합하여 총 256빈으로 구성된 특징벡터를 구하였다. 하지만 이러한 낮은 레벨의 특징 벡터들을 직접 분석하는 것은 많은 데이터들을 처리해야 하기 때문에 축소된 데이터 집합에서 처리하는 것이 바람직하며, 비록 축소된 데이터 집합을 사용하더라도 유사 영상들을 분류 하는데 필요한 효율적인 특징들은 추출할 수 있어야 한다.

주성분 분석은 이러한 데이터 특징들을 추출할 수 있는 수학적 기법으로서 결과 데이터의 차원과 데이터간의 연관관계를 줄여준다[9]. 이러한 축소는 데이터 분산의 대부분을 포함하고 있는 처음 몇 개의 주성분 (principal components)들을 선택함으로써 가능해진다. 본 연구에서는 이러한 주성분 분석을 적용하기 위해 먼저 각 요약 비디오 영상의 특징벡터를 세로 방향으로 나열하여 하나의 행렬을 만든다. 조사하는 모든 비디오들의 숫자를 N 이라고 할 때, 만들어지는 행렬의 가로 크기는 모든 요약 비디오 영상들의 숫자인 N 이 되고 세로 크기는 요약 비디오 영상의 특징벡터의 크기인 256이 된다. 행렬의 j 번째에 있는 $256-by-1$ 특징벡터를 f_j , 이들 특징 벡터들 f_1, f_2, \dots, f_N 의 평균벡터를 u , 또한 $\psi_j = f_j - u$ 라고 정의 할 때 이들 벡터 집단의 공분산 행렬 C_j 는 (3)과 같이 표현된다.

$$C_j = \frac{1}{N} \sum_{j=1}^N \phi_j \phi_j^T \quad (3)$$

C_j 의 256개의 정규 직교 고유벡터를 E_1, E_2, \dots, E_{256} 그에 상응하는 고유치들을 $\lambda_1 > \lambda_2 > \dots > \lambda_{256}$ 라고 할 때, 처음 K 개의 고유벡터 들의 선형조합은 대부분의 정보를 포함하는 저차원의 요약 비디오 영상 공간으로 스패ן(span)한다. 따라서 각 요약 비디오 영상의 특징벡터를 K 개의 가장 큰 고유치들에 대응하는 K 개의 고유벡터에 투영할 경우 다음 (4)에 의하여 자신의 고유영상 P_j 으로 변환된다.

$$p_j = E^T \phi_j = [p_{j1} p_{j2}, \dots, p_{jk}]^T \quad \text{단, } 1 \leq j \leq 256. \quad (4)$$

이때 벡터 P_j 의 요소들을 주성분들이라 부르며 j

번째 요약 비디오 영상의 간결한 정보를 포함한다. 요약 비디오 영상의 대각선 길이는 $\sqrt{M_1^2 + M_2^2}$ 이므로 이들을 k 개의 저차원 특징벡터로 투영하는 계산량은 $O(K * \sqrt{M_1^2 + M_2^2} + K * N)$ 이 된다.

3.3 군집화를 통한 유사 비디오 검출

저차원 특징 벡터 공간에 투영된 요약 비디오 영상들 중에서 유사한 요약 비디오 영상들은 비슷한 공간상에 위치하게 될 것이다. 따라서 본 연구에서는 유사한 비디오들을 구분하기 위하여 투영된 요약 비디오 영상들을 대상으로 군집화를 수행한다. 본 연구에서는 패턴공간에 주어진 유한개의 특징 벡터들을 자동으로 군집화 하기 위하여 평균-결합(average-linkage)에 기반을 둔 계층적인(hierarchical) 군집화 방법을 사용하였다[10]. 군집 C_i 는 i 번째 군집을 의미하며 n_i 멤버를, 군집 C_j 는 j 번째 군집을 의미하며 n_j 개의 멤버를 가졌다고 가정하고, 두 요약 비디오 영상의 특징벡터 f 와 f' 의 거리 $d(f, f')$ 는 유클리드 거리(Euclidean distance)라고 할 때 평균결합에 기반을 둔 계층적인 군집화 방식에서 두개의 군집 사이의 거리 $D(C_i, C_j)$ 는 다음 (5)와 같이 정의되며,

$$D(C_i, C_j) = \frac{1}{n_i n_j} \sum_{f \in C_i, f' \in C_j} d(f, f') \quad (5)$$

군집화 알고리즘은 다음과 같다.

1) 특징 벡터 공간에 투영된 N 개의 요약 비디오 영상 각각에 대하여 하나의 군집, 즉 $\{C_1\}, \{C_2\}, \dots, \{C_N\}$ 의 N 개의 군집을 만들며 군집변수에 n 을 설정한다.

2) 군집변수가 1이 되거나 $A \neq B$ 인 모든 군집에 대하여 만약 $D(A, B) > \theta$ 이면 중단하고 그렇지 않으면 (3)으로 간다.

3) 모든 군집 A, B 에 대하여 $D(E, F) \leq D(A, B)$ 를 만족하는 가장 가까운 두개의 군집 E 와 F 를 찾는다.

4) 군집 E 와 F 를 합하여 새로운 군집으로 만들며 군집변수를 하나 줄인 후 (2)번으로 간다.

이때 임계값 θ 는 어떤 두 개의 군집이 하나로 결합할 수 없는 최소 임계 값으로써, 이 임계 값에 의하여 최종의 유사 비디오 군집의 숫자가 결정된다. 평균 결합 군집화 기법의 복잡도는 $O(N^2)$ 에 가능하므로[10] 요약 비디오 영상을 사용해 유사 비디오들을

구한 전체의 복잡도는 $O(K * \sqrt{M_1^2 + M_2^2} + K * N + N)$ 이다.

4. 구현 및 결과

4.1 구현 환경

본 논문에서 구현된 시스템은 Intel Pentium III 800Mz, 256 Ram 상의 Microsoft Windows 2000 Professional 환경에서 구현되었으며, 전체적인 인터페이스와 구현은 Visual C++ 6.0을 사용하였다. 또한 비디오 데이터로부터 영상 정보를 직접 추출하기 위하여 DirectShow 9.0 SDK상에서 동영상 추출 알고리즘을 사용하였다.

웹상의 비디오 데이터는 대부분의 경우 사이즈가 작은 클립(clip)의 형태로 존재한다. 따라서 실험에 사용한 비디오들을 대략 2분에서 3분 정도의 크기로 주로 웹상에서 구한 CF, 뮤직, 뉴스, 드라마 등 다양한 내용들의 비디오들을 사용하였다. (그러나 본 연구에서 제안하는 방식은 일반 영화와 같은 사이즈가 큰 비디오에도 적용이 된다.) 제안하는 기법의 성능을 조사하기 위해 전체 60개의 비디오 클립들을 사용하였다.

각 비디오마다 개략적으로 3개의 유사 비디오들이 존재하기 때문에[5], 본 연구에서도 60개의 비디오들 중 서로 내용이 다른 비디오는 20개 그룹으로 설정하였으며 각 비디오 그룹은 3개의 유사 비디오들로 구성이 되어있다. 유사 비디오들은 내용은 같으나 프레임 비율(fps가 다른 것), 사이즈, 압축 방식, 포맷 등이 다른 비디오들로 구성하였다.

4.2 구현 결과 및 분석

제안하는 방식의 전체적인 성능을 조사하기 위하여 정보 검색에서 주로 사용하는 recall과 precision의 개념을 이용하여 실험 결과를 측정하였다. α 를 제안하는 방식에 의하여 유사하다고 검색된 비디오 그룹의 숫자, β 를 유사하다고 검색된 비디오들 중에서 실제로 유사한 비디오 그룹의 숫자, δ 를 조사하는 모든 비디오들 중에서 실제로 유사한 비디오 그룹의 숫자라고 정의 할 때 precision과 recall은 각각 (6), (7)과 같이 정의된다.

$$precision = \frac{\beta}{\alpha} \tag{6}$$

$$recall = \frac{\beta}{\delta} \tag{7}$$

제안한 방식으로 유사한 비디오들을 검출하기 위해서는 다음 3개의 인자 값들을 결정해야 한다. 비디오에서 시드 영상과 유사한 M 개의 프레임들을 추출할 때의 M 값, 이들 프레임들을 추출하기 위해 비디오를 L 개의 블록으로 나눌 때의 L 값, 요약 비디오 영상을 저차원 특징 벡터로 투영하기 위한 K 개의 주성분 숫자를 결정할 때의 K 값을 결정해야 한다. 이들 중 시드 영상과 유사한 M 개의 프레임 숫자는 사용하는 비디오의 대각선 길이에 의하여 정해지는 요약 비디오 영상의 세로 크기와 전체적인 크기를 고려하여 $M=128$ 값을 갖도록 하였다.

우선 요약 비디오 영상을 주성분 분석을 통하여 저차원 특징 벡터 공간으로 투영을 할 때 몇 개의

주성분들을 선택하는 것이 필요한지를 조사하였다. 주성분 차원수를 구하는 실험을 간단하게 하기 위하여 비디오 전체를 하나의 블록으로 사용하였다. 그림 3에 나타난 것처럼, precision과 recall의 비율은 주성분의 숫자가 증가할수록 높아지는 것을 알 수 있으며, 주성분의 숫자가 5일 경우 성능과 계산 복잡도의 균형을 맞춘 값을 가짐을 알 수 있다.

그림 4는 시드 영상과 유사한 프레임들을 선택할 때 비디오를 몇 개의 블록으로 나누는 것이 필요한지를 조사하였다. 이때 주성분의 숫자는 그림 3의 결과에 의해 5를 사용하였다.

그림 4를 통해 보듯이 주성분의 숫자가 5일 경우 비디오 블록을 나누는 숫자가 클수록 precision값과 recall값이 커지는 것을 볼 수 있다. 이때 비디오 블록의 수가 4일 경우가 성능과 계산 복잡도면에서 최적화된 값을 가짐을 알 수 있다.

표 1에서는 그림 4를 통해서 구한 성능과 검색 효율성을 만족시킬 수 있는 주성분 값 5와 비디오 블록

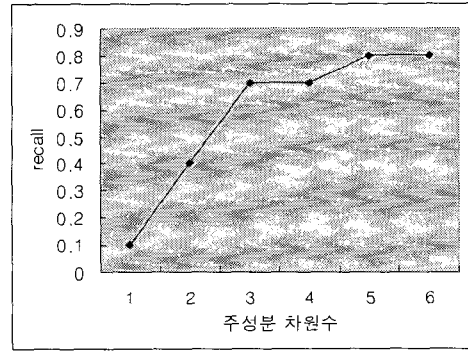
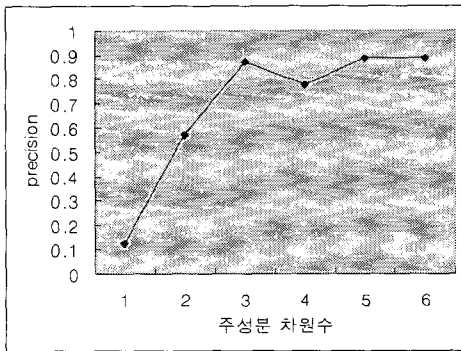


그림 3. 주성분 차원수에 따른 precision과 recall 값의 변화 그래프

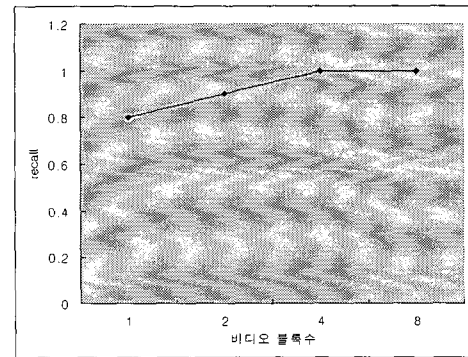
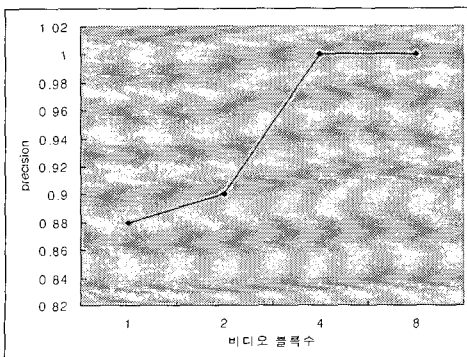


그림 4. 비디오 블록수에 따른 precision과 recall 값의 변화 그래프

표 1. 비디오 형식에 따른 유사성 검색 결과

	유사비디오 그룹 수	검출	오검출
different frame	20	18	2
Reformat	20	20	0
Editing	10	6	4
Total	50	44	6

수 4를 가지고 유사비디오 그룹들을 검출한 결과를 보인다. 이때 실험 데이터는 프레임 비율이 다른 2개의 유사 비디오들로 구성된 20개의 그룹, 서로 다른 포맷을 가지는 2개의 유사비디오들로 구성된 20개의 그룹, 편집을 통해 서로 다른 2개의 유사비디오로 구성된 10개의 그룹을 사용하였다.

표 1에서 보는 것처럼 검색 결과는 일부 장면 전환이 많은 데이터에서 일부 프레임 비율이 다른 경우나 인위적 편집에 의한 데이터 변경이 있는 경우에 오검출이 있는 것을 제외하고 대부분의 비디오 동영상의 경우 좋은 성능을 보이는 것을 알 수 있다.

그림 5에서는 표 1에 사용된 실험 데이터의 일부를 나타내고 있다. 그림 5의 (a)와 (b)에서는 원 영상에서 구한 요약 비디오 이미지, 프레임률이 틀린 유사비디오에서 구한 요약 비디오 이미지, 포맷이 틀린 유사비디오에서 구한 요약 비디오 이미지로 구성되어 보여주고 있다. 서로 다른 동영상의 요약 비디오 이미지는 서로 다름을 보여주고 있으며, 유사 비디오

의 요약 비디오 이미지는 서로 유사함을 보여주고 있다.

5. 결 론

웹상의 멀티미디어에 대한 관심의 꾸준한 증가와 전송 기술의 발달로 인해 실제 많은 수의 멀티미디어 데이터가 다양한 형식으로 존재하게 되었고, 그 데이터의 수는 갈수록 늘어날 것이다. 그중에서도 유사 비디오 데이터의 중복은 효율적인 비디오 데이터의 관리와 비디오 데이터 검색시 연속적인 정보 검색의 효율을 저하시키는 문제를 일으킨다. 때문에 유사 비디오들을 검색하여 미리 구분해 놓음으로써 정보 검색의 효율을 증가시켜 효율적인 데이터 검색을 가능하게 할 수 있다.

본 논문에서는 비디오 블록의 대표 프레임에서 추출한 영상 정보를 분석함으로써 유사 비디오 데이터를 검색해 내고 있다. 요약 비디오 영상을 추출하기 위해 시드 이미지와 영상의 위치정보를 고려한 특징 벡터를 추출하여 정확한 비디오 정보를 추출하며, 이를 주성분 분석을 통해 분석하여 비디오간의 관계를 구분하였다. 실험결과에 의하면 대부분의 서로 다른 프레임 비율을 가지는 경우나 비디오 데이터의 포맷이나 크기가 같은 경우에는 본 논문에서 제안한 유사 비디오의 검색 방법에 의해 좋은 결과가 나옴을 알 수 있다.

향후 연구에서는 비디오 블록을 나눌 때 일정한

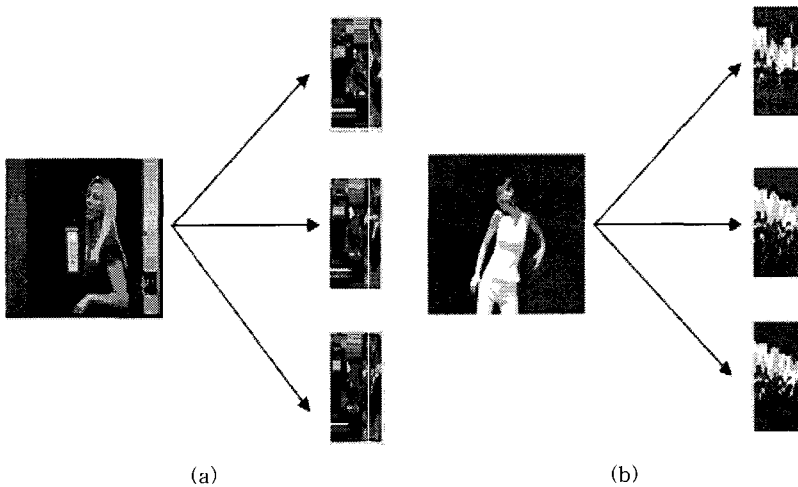


그림 5. 주성분 차원수가 5이고 비디오 블록수가 4인 경우의 요약 비디오 이미지

간격으로 나누는 것이 아니라 내용상 같은 부분으로 분할할 수 있는 방법을 연구하고, 이를 통해 더 효율적인 동영상 검색이 가능하도록 할 계획이다.

참 고 문 헌

[1] N. Shivakumar and H. Garcia-Molina, "Finding near-replicas of documents on the web," in *Proceedings of Workshop on Web Databases*. (WebDB '98), pp. 204-212. March 1998.

[2] R. Mohan and Y. Heights, "Video sequence matching," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 6, pp. 3697-3700, 1998.

[3] M. R. Naphade, M. M. Yeung, and B-L. Yeo, "A novel scheme for fast and efficient video sequence matching using compact signatures," *SPIE Conference on Storage and Retrieval for Media Databases 2000*, Vol. 3972, pp. 564-572, Jan 2000.

[4] D. A. Adjeroh, M. C. Lee, and I. King, "A distance measure for video sequence similarity matching," in *Proceedings International Workshop on Multi-Media Database Management Systems*, pp. 72-79, August 1998.

[5] S. C. Cheung and A. Zakhor, "Estimation of web video multiplicity," *Proc. SPIE Internet Imaging*, Vol. 3964, pp. 34-36, Jan. 2000.

[6] S. C. Cheung and A. Zakhor, "Efficient video similarity measurement and search," *ICIP 2000*, Vol. I, pp. 85-89, Sept. 2000.

[7] S. C. Cheung and A. Zakhor, "Video similarity detection with video signature clustering,"

Proceedings of the International Conference on Image Processing, Thessaloniki, Greece, Oct. 2001.

[8] H. Kim, et. al., *Visual rhythm and shot verification* Kluwer Academic Publishers, 2000.

[9] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes In C The Art of Scientific Computing*, CAMBRIDGE UNIVERSITY PRESS, United Kindgom, 1999.

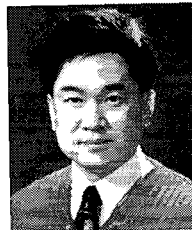
[10] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, NJ, USA, 1988.



유재만

2003년 광운대학교 컴퓨터과학
과 학사 졸업
2005년 광운대학교 컴퓨터공학
부 이학 석사
2005년~현재 (주) 다리테크 개
발 연구소 연구원

관심분야 : 멀티미디어 시스템, 영상/비디오 처리 및 패턴
인식



김우생

1985년 서울대 수료 및 Univer-
sity Texas at Austin 전
산학과 졸업
1987년 University of Minnesota
전산학 이학 석사
1991년 University of Minnesota
전산학 이학 박사

2001년 University of California at Berkeley 교환 교수
1992년~현재 광운대학교 컴퓨터 공학부 교수
관심분야 : 멀티미디어 시스템, 데이터베이스, 영상/비
오 처리 및 패턴인식