

# 연속하는 공간적 특징의 시간적 유사성 검출을 이용한 고속 동영상 검색

정회원 조 아 영\*, 양 원 근\*, 준회원 조 주 희\*, 임 예 은\*, 종신회원 정 동 석\*\*

## Fast Video Detection Using Temporal Similarity Extraction of Successive Spatial Features

Ayoung Cho\*, Won-Keun Yang\* *Regular Members*,  
Ju-Hee Cho\*, Ye-Eun Lim\* *Associate Members*, Dong-Seok Jeong\*\* *Lifelong Member*

### 요 약

멀티미디어 기술이 발전함에 따라 대용량의 데이터베이스의 관리와 불법 복제물 검출을 위한 동영상 검색의 필요성이 커지고 있다. 본 논문에서는 이러한 요구에 맞춰 대용량 데이터베이스에서 고속 동영상 검색을 수행할 수 있는 방법을 제안한다. 고속 동영상 검색 방법은 프레임의 휘도 분포를 이용하여 공간적 특징을 추출하고, 동영상의 시간적 유사성 지도를 생성하여 시간적 특징을 추출한다. 동영상의 공간적 특징과 시간적 특징을 식별자로 구성하고 단계적인 정합 방법을 수행한다. 실험에서는 원본 동영상과 밝기 변화, 압축률 변환, 자막/로고 삽입과 같은 다양한 변형을 이용하여 정확성, 추출 및 정합 속도, 식별자 크기를 측정하여 성능을 평가하였다. 또한, 제안한 방법의 파라미터를 실험적으로 선택한 과정을 기술하고 비교 알고리즘과 공간적 특징만을 이용한 단순 정합 결과를 제시하였다. 정확성, 검색 속도, 식별자 크기의 모든 결과에서, 제안한 고속 검색 방법이 대용량 데이터베이스의 동영상 검색에 가장 적합한 기술임을 보였다.

**Key Words** : Video Signature, Spatial Feature, Temporal Feature, Video Copy Detection

### ABSTRACT

The growth of multimedia technology forces the development of video detection for large database management and illegal copy detection. To meet this demand, this paper proposes a fast video detection method to apply to a large database. The fast video detection algorithm uses spatial features using the gray value distribution from frames and temporal features using the temporal similarity map. We form the video signature using the extracted spatial feature and temporal feature, and carry out a stepwise matching method. The performance was evaluated by accuracy, extraction and matching time, and signature size using the original videos and their modified versions such as brightness change, lossy compression, text/logo overlay. We show empirical parameter selection and the experimental results for the simple matching method using only spatial feature and compare the results with existing algorithms. According to the experimental results, the proposed method has good performance in accuracy, processing time, and signature size. Therefore, the proposed fast detection algorithm is suitable for video detection with the large database.

※ 이 논문은 인하대학교 교내학술연구비의 지원을 받아 수행된 연구임

\* 인하대학교 전자공학과 멀티미디어 연구실({ayoung, aida, noir.tama, swans2000}@inha.edu)

\*\* 인하대학교 전자공학과 교수 (dsjeong@inha.ac.kr)

논문번호 : KICS2010-09-436, 접수일자 : 2010년 9월 6일, 최종논문접수일자 : 2010년 10월 21일

## I. 서 론

디지털 콘텐츠의 급격한 증가로 대용량 데이터베이스를 효과적이고 효율적으로 관리할 수 있는 기술이 요구되고 있으며 실제 많은 기술의 개발도 이뤄지고 있다. 기본적으로 필요한 기술은 대용량 데이터베이스에서 사용자가 원하는 콘텐츠만을 걸러내어 출력하는 것으로, 많은 시스템에서 이러한 기술을 구현되어 상용화되고 있다. 이와 같은 콘텐츠 검색 기술들의 발전 과정을 보면, 과거에는 문서 검색을 위한 텍스트 검색 기술을 중심으로 개발되었다. 그러나 근래에는 네트워크 기술과 저장 장치가 발전됨에 따라 전송과 배포가 쉬워지면서 용량이 큰 정지영상 및 동영상 검색하는 기술도 함께 부각되고 있다. 현재 주로 사용되는 방법 중의 하나는 정지영상이나 동영상에 키워드를 삽입시키고 그 키워드를 기준으로 검색하는 방법이다. 그러나 키워드의 부재나 제거로 인해 검색이 불가능한 경우와 주관적이거나 의미론적 표현의 모호성으로 인한 한계에 직면하게 되면서, 그 대안으로 영상 자체의 정보를 사용하는 내용기반(content-based) 검색 방법이 제시되고 있다. 예를 들어, 사용자는 원하는 동영상을 얻기 위해 질의 동영상을 사용하고 검색 시스템은 질의 정보와 밀접한 관련성을 갖고 있는 동영상들을 결과로 출력하는 방법이다.

이러한 내용기반 검색은 대용량 데이터베이스의 자동 관리에도 응용될 수 있다. 내용기반 검색을 이용하면 관련 데이터의 분류, 데이터의 색인 및 정렬, 불필요하게 공간을 차지하는 중복 데이터의 관리 및 삭제 등을 수행할 수 있다. 다른 응용 분야로는 디지털 콘텐츠의 발달과 함께 대두되고 있는 문제점인 불법 복제물 추적이 있다. 불법 복제에는 데이터의 제목 및 키워드를 변형시켜 배포하는 것은 물론이고, 데이터를 사용하는데 문제가 되지 않을 정도의 변형을 가하는 경우가 있다. 그리고 특별히 동영상의 경우에는 전체를 구간별로 자른 형태가 유포되기도 한다. 그러므로 질의 콘텐츠의 내용을 기반으로 변형된 콘텐츠를 찾거나 동영상의 경우에는 일부 동일한 구간을 갖는 동영상을 찾아내는 기술로 내용기반 검색이 적합한 방법이다.

동영상은 데이터의 크기가 크고 다양한 정보를 담고 있기 때문에 동영상 내용기반 검색 기술 개발은 어려운 과제이지만 그 필요성이 나날이 커지면서 여러 기술이 제안되고 있다. 이러한 검색 기술은 기본적으로 높은 정확성과 빠른 속도가 필수적으로

요구된다. 먼저 정확성은 동일한 데이터는 물론이고 변형이 일어나더라도 검색할 수 있는 능력을 말한다. 동영상은 재생되는 플레이어에 따라 인코딩이 다시 수행 되거나, 저장장치나 전송환경의 제한된 용량을 맞추기 위한 목적으로 다양한 방법으로 변형된다. 그러므로 높은 정확성을 확보하기 위해서는 화질의 열화나 밝기 변화, 해상도 변환, 자막 추가 등이 일어난 변형된 동영상과 같이 원본 데이터와 완전히 동일하지 않은 데이터도 동일 데이터로 구분 지을 수 있어야 한다. 여기서 의미하는 정확성은 서로 다른 동영상은 검색 결과에서 제외시켜야 한다는 뜻도 내포하고 있다. 그러므로 변형으로 인해 발생하는 특징의 차이와 서로 다른 데이터 사이의 특징 차이를 구분할 수 있는 기술이 요구된다. 그리고 빠른 검색 요건은 아무리 검색 결과가 정확하더라도 속도가 느리면 사용자들은 검색 시스템을 사용하는 것을 꺼리게 되기 때문에 필수적이다. 정확도를 높이기 위해 정밀한 식별자를 추출하고 정합하는 방법을 사용할수록 높은 복잡도를 요구하고 이는 검색 속도의 저하를 야기한다. 따라서 정밀한 기술의 사용으로 정확도가 올라가는 것과 비교해서 저하되는 검색 속도가 용납할 만큼이어야 실제 환경에 적용할 수 있다.

초기의 동영상 검색 기술은 동영상을 정지영상의 나열로 바라보고 정지영상 검색 기술을 확장하는 방법이 제안되었다. 그러나 모든 프레임은 정지영상으로 놓고 정지영상 검색 기술을 그대로 적용하여 식별자를 추출하기에는 추출과 정합 시간이 길고, 식별자 크기가 크다는 문제가 발생한다. 그 해결책이 크게 두 가지 방향으로 제시되고 있는데, 하나는 모든 프레임에서 추출해도 될 만큼 추출 방법이 간단하고 추출 결과의 크기도 작은 방법이고, 다른 하나는 샷(shot) 검출과 키프레임 추출 기술 등을 사용하여 몇 개의 특정 프레임에서만 식별자를 추출하는 방법이다. 전자는 작은 특징만을 추출할 수 있기 때문에 식별자의 강인성이 매우 요구되고, 후자는 식별자의 강인성뿐 아니라 샷 검출 및 키프레임 추출의 반복성이 요구된다.

Julien Law-To 등<sup>[1]</sup>은 다양한 동영상 복제검출 기술의 비교 연구를 보고하였다. 이 논문에서는 기술을 전역(global) 식별 방법과 지역(local) 식별 방법으로 분류하고 각각의 성능을 비교하여 평가하였다. 전역 식별 방법에는 칼라 히스토그램과 같은 특징을 이용한 방법도 많이 제시되고 있지만, 가장 대표적인 방법은 순위 척도(ordinal measure)를 적용

한 방법이 있다. 순위 척도는 정지영상의 특징 추출을 위해 제안된 기술로서 그 연산이 간단하면서 동시에 강인한 성능을 보여주고 있기 때문에 많은 논문에서 이를 응용하여 동영상 검색 기술을 . Hua 등<sup>[2]</sup>은 3x3 순위 척도를 이용해 특징을 추출하고 단계적 정합 방법을 사용하여 동영상 검출을 수행하였다. 그리고 Kim과 Vasudev<sup>[3]</sup>은 2x2 순위 척도를 사용한 공간적 특징과 시간적 변화 특징을 결합하여 동영상 검색에 적용하였고 Chen 등<sup>[4]</sup>은 2x2 순위 척도의 시간적 특징을 이용하였다. 지역 식별 방법은 특징점(interest point) 검출하고 그 특징점을 중심으로 영역 특징을 추출하여 동영상 검색에 적용한 방법이 대표적이다<sup>[5],[6]</sup>. 이러한 방법은 동영상의 기하학적인 변형에 강인하면서 객체를 중심으로 동영상을 구별하기 위해 사용된다. 지역 식별 방법은 연산이 복잡하면서 특징점 마다 영역 특징을 추출하기 때문에 최종 추출 식별자의 크기가 특징점수에 의해 급격히 증가하는 단점이 있다. 그러므로 키프레임 기술을 접목하여 특징을 추출하는 프레임수를 줄이거나, 추출 데이터의 크기를 축소하기 위한 후처리 방법을 적용하는 방안을 제시하여 기술의 단점을 보완하여 제안되고 있다.

최근에 정지영상 및 동영상 검색을 위해 부각되고 있는 또 다른 기술은 BoW(Bag of Words)이다<sup>[7],[8]</sup>. 문서 분석을 위해 개발된 BoW 기술의 개념을 영상에 접목시켜서 검색에 응용하는 기술이 제안되고 있다. 이 기술은 추출된 특징을 클러스터링을 이용해 시각단어(visual words)로 만들고 각각의 시각단어의 발생빈도를 특징으로 이용하여 동영상을 정합하는 방법이다. 시각단어를 구성하기 위해 주로 SIFT(Scale Invariant Feature Transform)<sup>[9]</sup>나 SURF(Speeded Up Robust Features)<sup>[10]</sup>의 지역 특징을 이용한 방법이 제안되었다. 즉, BoW 기술은 위의 지역 식별 방법과 결합하여 지역 특징을 시각단어의 특징으로 재구성 시켜서 특징의 크기를 축소하고 정합 속도를 빠르게 한다. BoW를 이용한 기술로는 자동차, 인물, 자전거 등의 객체 분류 및 객체 중심의 검색 등을 위한 기술들이 있다.

대용량 데이터베이스에서 중복 제거 및 동일 데이터의 관리와 불법 동영상 추적과 같은 응용에서는 원본의 정보가 크게 손상되지 않은 데이터의 검색을 수행하면 된다. 즉, 압축 방식의 변화 및 용량 제한으로 인한 변형 등이 가해진 동영상을 검색할 수 있어야 하며, 이러한 변형이 가해진 동영상을 대

용량 데이터베이스에서 고속 검색을 수행할 수 있어야 한다. 그러므로 본 논문에서는 전역 식별 방법으로 이러한 변형에 강인한 고속 동영상 검색 방법을 제안한다. 제안된 방법은 강인성과 구별성이 높은 공간적 특징을 추출하고, 검색 속도를 높이기 위해 시간적 유사성 지도를 사용하여 동영상의 시간적 특징을 추출한다. 추출된 특징을 식별자로 구성하고 단계적인 정합 방법에 의해 고속 정합을 수행한다. 본 논문의 구성은 다음과 같다. II장에서는 공간적 특징과 시간적 특징을 사용한 고속 동영상 검색 방법을 기술한다. III장에서는 실험을 통해 제안된 알고리즘이 일반적인 변형에서 강인하면서 빠른 검색이 가능함을 보여주고, 마지막으로 IV장에서는 결론을 맺는다.

## II. 동영상 검색 방법

본 논문에서 제안하는 동영상 검색 방법은 휘도의 분포를 이용한 공간적 특징과 이웃한 프레임 간의 공간적 특징의 변화를 측정된 시간적 특징을 사용한다. 그림 1은 전체 시스템의 흐름과 함께 추출되는 특징들을 보여준다. 입력된 동영상은 먼저 해상도와 프레임률을 고정된 값으로 정규화 시킨다. 프레임률 정규화를 통해 동영상의 프레임 수는 N이 되고 N개의 정규화된 모든 프레임에서 공간적 특징을 추출한다. 다음으로 프레임 간의 공간적 특징의 차이를 계산하여 시간적 유사성 지도를 생성하고, 시간적 유사성 지도의 분포에 따라 컷을 검출한다. 이렇게 검출된 컷의 개수와 컷의 위치를 나타내는 색인 비트열이 시간적 특징으로 추출된다. 각 과정에 대한 자세한 설명은 다음과 같다.

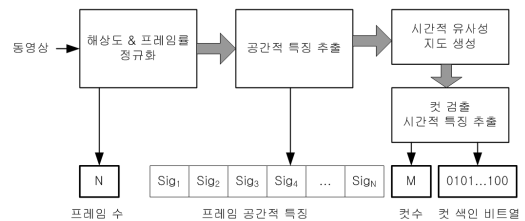


그림 1. 공간적 특징과 시간적 특징을 이용한 동영상 식별자 추출 과정

Fig. 1. Video signature extraction using spatial and temporal features

2.1 동영상 정규화

먼저 동영상에서 특징을 추출하기 전에 영상의 크기와 프레임률(fps)을 정규화한다.

이러한 과정으로 스케일 변환 및 프레임률 변환에도 불변한 특징을 추출할 수 있다. 동영상이 다양한 변형으로 정보가 손실되거나 손상되더라도 보통 동영상의 해상도와 프레임률 정보를 얻을 수 있다. 그 정보를 이용해서 영상의 크기는 256x256으로, 프레임률은 4fps로 정규화한다.

2.2 휘도 분포를 이용한 공간적 특징

공간적 특징 추출은 동영상의 컬러 정보가 아닌 흑백 정보를 이용한다. 256x256 크기로 정규화된 각 프레임은 흑백 영상으로 변환하고, 영상의 중심을 기준으로 반지름 128의 원의 내부 영역에서 특징을 추출한다. 그림 2의 좌측과 같이 원의 내부를 반지름과 각도를 균등 분할하고, 분할된 각 소구역의 대표값으로 평균 화소값을 계산한다. 동영상은 주로 중심 영역에 의미 있는 객체를 담게 되므로, 이와 같은 영역 분할은 중심에 가까운 영역일수록 세밀한 정보 이용하게 되어 구별성을 높일 수 있다. 이 영역 분할은 평면직교좌표계(Cartesian coordinates)를 극좌표계(polar coordinates)로 변환하면 블록의 형태로 간편하게 수행할 수 있다. 좌표계 변환은 식 (1)을 이용하여 (x, y) 좌표를 (r, θ) 좌표로 변환하며, 여기서 (x, y)의 원점은 영상의 중앙에 위치한다.

$$\begin{aligned}
 x &= r \cos \theta, \quad y = r \sin \theta, \\
 r &= \sqrt{x^2 + y^2}, \\
 \theta &= \begin{cases} \arctan(y/x) & \text{if } x > 0 \text{ and } y \geq 0 \\ \arctan(y/x) + 2\pi & \text{if } x > 0 \text{ and } y < 0 \\ \arctan(y/x) + \pi & \text{if } x < 0 \\ \pi/2 & \text{if } x = 0 \text{ and } y > 0 \\ 3\pi/2 & \text{if } x = 0 \text{ and } y < 0 \end{cases} \quad (1)
 \end{aligned}$$

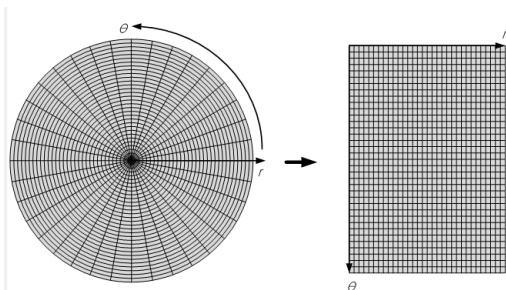


그림 2. 직교좌표계의 극좌표계 변환  
Fig. 2. Conversion from Cartesian coordinates to polar coordinates

변환된 극좌표계에서 그림 2의 우측과 같이 반지름은 D 단계, 각도는 A 단계의 격자 형태의 소구역으로 균등 분할한다. 각 소구역의 평균 화소값을 대표값으로 하면, 결국 DxA 크기의 영상을 구할 수 있다. 본 논문에서는 D=32와 A=36을 사용하여 32x36의 극좌표계 영상을 얻게 된다.

공간적 특징 추출 방법은 동심원 기반 알고리즘을 적용한다<sup>[11]</sup>. [11]에서는 반지름을 균등 분할한 고리 영역을 단위로 특징값을 추출한 반면, 본 논문에서는 각도를 균등 분할한 부채꼴 영역을 단위로 하여 특징값을 추출한다. 동영상은 특성상 영상의 외곽의 정보가 손실될 가능성이 크므로 부채꼴 모양으로 분할된 영역에서 특징값을 계산하여 손실에 대한 영향을 완화시킬 수 있다. 특징의 종류는 부채꼴 영역 a의 평균 화소값 SA(a), 근접 소구역 절대차 평균값 SD(a), 분산값 SV(a)으로 크게 3 종류이다. 각 특징을 계산하는 방법은 식 (2)-(4)와 같다. (1 ≤ a ≤ A)

$$SA(a) = \frac{1}{D} \sum_{d=1}^D s(d, a), \quad (2)$$

$$SD(a) = \frac{1}{D} \sum_{d=2}^D |s(d, a) - s(d, a-1)|, \quad (3)$$

$$SV(a) = \frac{1}{D} \sum_{d=1}^D \{s(d, a) - SA(a)\}^2. \quad (4)$$

여기서 s(d, a)는 그림 2의 DxA 영상에서 해당 좌표의 값을 가리킨다. 구별성을 높이기 위해 위에서 구한 3 종류의 특징값으로 식 (5)와 같이 이웃한 부채꼴 영역과의 1차 변화량을 추가적으로 구한다. 그리고 식 (6)와 같이 1차 변화량에 대한 변화량, 즉 2차 변화량을 계산하여 특징으로 추가한다. 단, SV의 경우에는 그 특징 자체가 변화량을 측정하므로 SV의 2차 변화량은 사용하지 않는다. 따라서 총 8가지의 특징값을 계산하게 된다.

$$\begin{aligned}
 V_{SA}^1(a) &= SA(a+1) - SA(a), \\
 V_{SD}^1(a) &= SD(a+1) - SD(a), \\
 V_{SV}^1(a) &= SV(a+1) - SV(a), \\
 &\quad (1 \leq a \leq A-1)
 \end{aligned} \quad (5)$$

$$\begin{aligned}
 V_{SA}^2(a) &= V_{SA}^1(a+1) - V_{SA}^1(a), \\
 V_{SD}^2(a) &= V_{SD}^1(a+1) - V_{SD}^1(a), \\
 &\quad (1 \leq a \leq A-2)
 \end{aligned} \quad (6)$$

이와 같이 추출된 특징은 해싱(hashing)을 통해

이진수로 변환된다. 수치로 나타낸 특징값은 영상의 변화로 인해 미세하게 값이 변화될 수 있다. 이러한 변화에는 둔감하면서 데이터 크기를 줄일 수 있기 때문에 많은 알고리즘에서 해싱을 사용하고 있다. 위에서 추출하한 8가지 특징값을 각각 특징 별로 부채꼴 영역 순서대로 나열하고 해싱을 수행한다. 해싱은 식 (7)과 같이 간단히 크기 비교에 의해 계산된다.

$$H(a) = \begin{cases} 1, & \text{if } F(a+1) > F(a) \\ 0, & \text{o.w.} \end{cases} \quad (7)$$

변화량의 계산 과정이나 해싱을 통해 각 특징마다 값의 개수가 1에서 3씩 감소하므로 36 레벨의 각도를 분할한 영역에서 추출한 최종 데이터의 크기는 한 프레임 당 273-bit이다.

### 2.3 시간적 유사성 지도를 이용한 시간적 특징

동영상에 변형이 일어나면 그 공간적 특징도 함께 변하기 때문에 원본 동영상과 변형 동영상을 비교하면 다소 차이가 발생하게 된다. 그러나 일반적으로 변형은 동영상 전체에 동일하게 영향을 미치기 때문에 동영상 안에서 이웃한 프레임과의 관계는 변형에 의해 크게 달라지지 않는다. 본 논문에서는 이러한 특성을 고려해서 시간적 유사성 지도를 만들고 이 안에서 동영상 자체의 시간적 변화를 측정하여 컷 검출을 통해 시간적 특징을 추출한다.

시간적 유사성 지도는 앞에서 추출한 프레임 간의 공간적 특징을 이용하여 만들 수 있다. 시간적 유사성 지도는 원본 동영상의 모든 프레임의 공간적 특징을 서로 비교하여 그 차이값으로 표현한다. 즉, 프레임의 수가 N이라면 NxN의 시간적 유사성 지도가 생성된다.

그림 3은 차이값을 화소값으로 표현한 시간적 유사성 지도를 보여준다. 그림 3의 좌측은 5분 37초의 원본 동영상을 4fps의 정규화 프레임률로 공간적 특징을 추출하고, 그 공간적 특징들 간에 차이값을 이용하여 구성한 시간적 유사성 지도를 보여준다. 그리고 중앙과 우측은 원본 동영상에서 60초 이후부터 1분 길이의 클립을 잘라내고 밝기 변화와 자막/로고 삽입 변형을 가한 후에, 같은 방법으로 생성한 시간적 유사성 지도를 보여준다. 원본 동영상의 시간적 유사성 지도에서 흰색 정사각형으로 표시한 부분은 잘라낸 1분 클립에 해당하는 정보가 되고 변형된 1분 클립과 정합이 되는 영역이다.

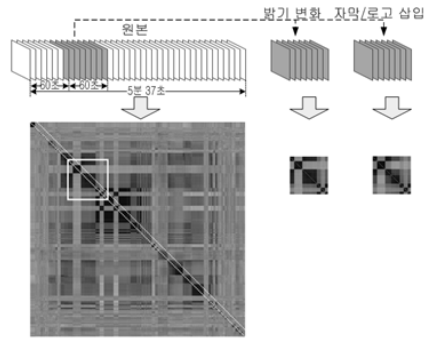


그림 3. 원본 동영상, 밝기 변화 클립, 자막/로고 삽입 클립의 시간적 유사성 지도  
Fig. 3. Temporal similarity map of original video, brightness changed clip, text/logo overlaid clip

시간적 유사성 지도에서 동영상의 시간적 특징을 추출하기 전에 시간적 유사성 지도의 이진화를 수행한다. 동영상의 장면 전환은 평균 6.5초에서 발생하는 것을 고려하여<sup>[13]</sup>, 현재 프레임에서 10초 이내에 해당하는 영역의 평균값을 구하여 이진화를 위한 임계치로 사용한다. 정규화 프레임률이 4fps이므로 그림 3 좌측에 점선으로 나타낸 것처럼 대각선 축으로부터 너비 40내에 해당하는 영역의 평균을 구한다. 이진화는 임계치보다 작으면 1, 그렇지 않으면 0으로 나타낸다.

다음은 시간적 특징을 추출하기 위한 과정으로 컷 검출을 수행한다. 이진화된 시간적 유사성 지도에서 대각선에 위치한 픽셀을 기준으로 X축 양의 방향으로 이동하면서 1에서 0으로 변경되는 위치를 표시한다. 같은 x좌표를 가지는 라인 위에 표시된 픽셀의 수가 4(1초)를 초과하면, 이 위치에 해당하는 프레임에서 컷이 발생했다고 판단한다. 만약 거리가 2 이하의 두 프레임에서 모두 컷이 발생되었다고 판단되면 표시된 픽셀의 수가 더 많은 위치에 해당하는 프레임만을 컷으로 판단한다. 그러므로 이웃한 컷은 최소한 3 이상 떨어져 있게 된다. 다음은 컷의 위치를 표시하는 컷 색인 비트열을 생성한다. 이 비트열은 N개의 프레임에서 컷의 위치를 나타내기 위해 N-bit로 표현되고 비트열에서 컷이 발생하는 위치에 해당하는 비트는 1로 표시한다. 즉, M개의 컷이 검출되면 비트열에서 1로 표시된 비트는 M개가 된다.

### 2.4 동영상 검색을 위한 식별자의 구성

공간적 특징과 시간적 특징의 추출을 마치면 동영상 검색에 필요한 식별자는 다음과 같이 구성된다.



컷수 | 컷 색인 비트열 | M개의 키프레임 공간적 특징

그림 4. 식별자의 구성  
Fig. 4. Composition of Signature

다. 시간적 특징인 컷의 개수와 컷 색인 비트열을 식별자로 사용하고, 공간적 특징은 컷의 위치에 해당하는 프레임에 키프레임으로 선택하여 해당 프레임의 공간적 특징만을 식별자로 사용한다. 여기서 키프레임은 컷의 위치가 변형으로 인해 흔들릴 수 있는 것을 고려하여 컷으로부터 -0.5초에 해당하는 프레임을 선택한다. 결과적으로 식별자는 그림 4와 같이 컷 수 M, 컷 색인 비트열, 키프레임의 공간적 특징열로 구성된다.

2.5 식별자의 단계적 정합 방법

본 논문에서 질의 동영상은 원본 동영상의 서브 동영상이라는 가정으로 질의 클립이라고 명하고, 그러므로 질의 클립이 항상 원본 동영상 보다 짧다고 가정할 수 있다. 동영상에서 단순 정합 방법은 짧은 질의 클립이 원본 동영상을 처음부터 스캔하면서 가장 잘 정합되는 위치를 찾아나간다. 다시 말하면, 원본 동영상의 프레임에서 추출한 공간적 특징의 나열과 질의 클립의 프레임에서 추출한 공간적 특징의 나열을 비교하면서 유사도가 가장 큰 위치를 찾아나간다. 하지만 이 방법을 수행하려면 대략 원본 동영상과 질의 클립의 프레임 수의 곱의 계산이 수행되기 때문에 정합 속도가 매우 느리고 효율성이 떨어지게 된다.

본 논문에서는 제안한 식별자를 가지고 고속 정합을 하기 위해 단계적인 정합 방법을 제시한다. 먼저 두 식별자에서 시간적 특징인 컷 개수와 컷 색인 비트열을 차례대로 비교한다. 컷 개수 비교는 극심한 차이가 나는 동영상을 걸러내기 위해 사용된다. 원본 동영상의 컷 개수가 질의 클립의 개수의 절반도 되지 않는다면, 이 동영상은 서로 완전히 다르다고 판단하고 더 이상 비교하지 않는다. 다음으로 컷 색인 비트열은 원본 동영상의 비트열 *ori\_BS* 위를 질의 클립의 비트열 *query\_BS*로 스캔하면서 가장 유사성이 높은 위치를 찾기 위해 사용된다. 질의 클립의 비트열 *query\_BS*와 비교하고자 하는 원본 동영상의 서브 비트열 *sub\_ori\_BS* 사이에서 동일 위치에 존재하는 컷의 수 *num\_pair*를 측정하고, 아래와 같은 식으로 유사성 점수 *S*를 계산한다.

$$S(query\_BS, sub\_ori\_BS) = 2 * num\_pair / (num\_sub\_ori\_cut + num\_query\_cut) \quad (8)$$

여기에서 *num\_sub\_ori\_cut*은 원본 서브 비트열에서의 1의 개수이고 *num\_query\_cut*은 질의 클립 비트열에서의 1의 개수이다. 질의 클립의 비트열이 원본 동영상의 비트열에서 가장 잘 부합되는 위치가 결정되면, 그 위치와 유사성 점수를 이용해 정합의 다음 단계로 넘어간다. 유사성 점수가 임계치 *Th1* 보다 크면 두 동영상은 서로 연관성이 존재할 확률이 높기 때문에 부합되는 위치에서 공간적 특징을 비교하고, 그렇지 않으면 두 동영상 사이에 관련이 없는 것으로 결정한다.

키프레임 공간적 특징열의 비교는 두 동영상의 컷 색인 비트열에서 동시에 1인 비트에 해당하는 키프레임의 공간적 특징을 가지고 해밍거리(Hamming distance)를 계산한다. 이 평균 거리가 임계치 *Th2* 보다 작으면 최종적으로 원본 동영상과 질의 클립이 부합된다고 판단한다.

III. 실험 및 결과

3.1 실험 환경

실험에서는 서로 다른 두 종류의 데이터베이스를 구성한다. 첫 번째 데이터베이스(DB1)는 3분 길이의 원본 동영상과 1분 길이의 다양한 변형이 가해진 질의 클립을 사용하며, 제안하는 방법에서 사용되는 파라미터를 선택하기 위해 이용된다. 이 데이터베이스에는 웹에서 다운받은 198개의 영화, 뉴스, 광고, 애니메이션 등의 다양한 원본 동영상이 포함되어 있다. 그리고 원본 동영상에서 시작 60초 이후부터 1분 클립을 잘라내고 다양한 변형을 가해서 질의 클립으로 구성한다. 여기에서 변형은 밝기 변화, 프레임률 변환, 흑백 변환, 해상도 축소, 압축률 변환, 자막/로고 삽입을 이용하였다. 두 번째는 성능 측정을 위한 데이터베이스(DB2)로서, 공정한 평가를 위해서 DB1과는 다른 데이터베이스를 이용하고 변형의 강도와 종류에도 변화를 주었다. DB2의 원본 데이터베이스는 “The Open Video Project”<sup>[12]</sup>에서 제공하고 있는 동영상 중에서 선택하였다. 이 동영상들은 연구를 위해 무료로 제공되고 있으며 해상도와 포맷이 다양한 동영상을 포함하고 있다. 본 논문에서는 4분 30초에서 6분 이내 길이의 동영상을 102개 선택하여 원본 동영상으로 사용하였다. 선택된 동영상은 29.97fps에 352x240, 320x240의

표 1. 동영상 변형의 종류와 강도  
Table 1. Types and strength of video modifications

변형종류 \ 레벨	Light	Medium	Heavy
밝기 변화	+9	-18	+36
대비 변화	1.1	0.9	N/A
프레임률 변환	25 fps	15 fps	N/A
자막/로고 삽입	10%	20%	N/A
테두리 잘림	15%	N/A	N/A
압축률 변환	min(1/2, 256Kbps)	min(1/4, 128Kbps)	N/A
해상도 축소	면적 1/2 축소	면적 1/4 축소	N/A
흑백 변환	Color to Gray	N/A	N/A

해상도를 가지며, 압축 포맷은 mpeg1과 mpeg4이다. 그리고 질의 클립은 첫 번째 데이터베이스와 동일한 방법으로 1분 클립을 생성하고 표 1과 같은 15 가지의 변형을 가한 데이터로 구축된다.

동영상 검색의 성능은 정확성과 수행 속도, 그리고 식별자의 크기를 측정하여 평가된다. 정확성은 질의 클립으로 원본 데이터베이스에서 자신의 원본 동영상과 부합하고 질의 클립의 위치까지 찾아내는 검출률(detection rate), 원본 동영상과 부합하지만 질의 클립의 위치를 잘못 찾아내는 위치 오류율(false position rate), 그리고 서로 다른 동영상을 원본 동영상으로 잘못 인식한 긍정 오류율(false positive rate)을 이용하여 평가된다. 이 평가로 동영상 검색을 위한 식별자의 강인성을 확인 할 수 있고, 특히 긍정 오류율을 통해 식별자의 구별성도 확인할 수 있다. 여기에서 원본 동영상에서 질의 클립의 위치 검색의 성공 여부는 0.5초의 오차 여유를 주고 판단한다. 그리고 데이터베이스의 일부 데이터를 가지고 식별자 추출과 정합 속도, 식별자 크기를 측정한다.

### 3.2 파라미터 구하기

제안된 방법의 파라미터 중에서 정규화 프레임률, 컷 색인 비트열의 정합 임계치 Th1 그리고 키프레임 공간적 특징열의 정합 임계치 Th2는 DB1을 이용하여 실험적으로 결정된다. 이후의 실험에는 이 값이 고정되어 성능 측정에 사용된다.

각 실험은 파라미터 값을 변경하면서 검출률을 측정하는 방법으로 수행된다. 정규화 프레임률은 Th1과 Th2가 각각 0.27, 83일 때와 0.20, 64일 때에

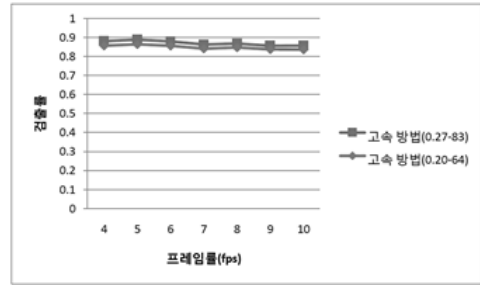


그림 5. 프레임률 정규화 값에 따른 검출률 비교  
Fig. 5. Detection rate by frame rate

표 2. 컷 색인 비트열의 정합 임계치에 따른 검출률 비교  
Table 2. Detection rate by matching threshold of cut index bitstream

Th1	0.25	0.26	<b>0.27</b>	0.28	0.29
검출률	0.858	0.858	<b>0.858</b>	0.857	0.856
Th1	0.30	0.31	0.32	0.33	0.34
검출률	0.855	0.854	0.852	0.852	0.845

대해 4fps부터 10fps까지 프레임률에 대한 검출률을 측정하였고 그 결과를 그림 5에서 보여준다. 실험 결과를 보면 4fps와 5fps에서 비교적 높은 검출률을 보이고, 그 이상으로 프레임률이 커지면 다소 검출률이 낮아지는 것을 볼 수 있지만 그 차이 또한 그리 크지 않음을 볼 수 있다. 그러므로 추출 속도를 위해 4fps의 정규화 프레임률을 선택한다.

다음으로 컷 색인 비트열의 정합 임계치를 0.25부터 0.34까지 증가시켜가면서 검출률을 측정하였다. 표 2의 결과를 보면 컷 정합 임계치가 0.28일 때부터 검출률이 감소하는 것을 볼 수 있다. 그러므로 최대 허용치인 0.27을 선택한다.

그리고 키프레임 공간적 특징열의 정합 임계치를 변경하면서 검출률을 측정하였고, 그 결과는 표 3과 같다. 검출률을 높이기 위해 임계치를 높게 선택할 경우 서로 다른 데이터를 검출하는 긍정 오류율이 커지므로 임계치 83을 선택한다.

표 3. 키프레임 공간적 특징열의 정합 임계치에 따른 검출률 비교  
Table 3. Detection rate by matching threshold of keyframe spatial features

Th2	70	75	80	81
검출률	0.870	0.873	0.878	0.879
Th2	82	<b>83</b>	84	85
검출률	0.880	<b>0.881</b>	0.881	0.881

### 3.3 성능 평가

비교 알고리즘은 전역 식별 특징을 이용한 비디오 검색의 가장 대표적인 두 가지 방법을 선택하였다. 두 방법은 2x2 순위 척도를 가지고 공간적 특징을 이용한 Hua 방법<sup>[2]</sup>과 3x3 순위 척도를 가지고 공간적 특징과 시간적 특징을 이용한 Kim 방법<sup>[3]</sup>이다. 이 두 가지 방법은 정합 과정에서 기본적으로 원본 동영상의 특징을 질의 클립의 특징으로 스캔하면서 검색하는 단순 정합 방법을 사용하고 있다. 그러므로 제안한 고속 검색 방법의 성능을 평가하기 전에 유사한 조건에서 성능을 비교하기 위해, 제안한 방법의 공간적 특징을 가지고 단순 정합 방법으로 그 성능을 측정한다.

성능 측정은 DB2를 사용하여 각 변형마다 검출률, 위치 오류율, 긍정 오류율을 측정하였으며 그 결과는 표 4와 같다. 실험 결과를 보면 모든 방법에서 다소 긍정 오류가 발생하는 것을 볼 수 있다. 이것은 데이터베이스에서 두 쌍의 유사한 동영상이 포함되어 있기 때문이다. 그리고 Kim 방법에서는 프레임률 정규화 과정이 없기 때문에 프레임률 변환에 대한 검출률은 0이 나온다. 그 밖의 변형에서는 비교적 검출률이 높아 보이지만 긍정 오류율이 매우 높아 식별자의 구별성이 떨어짐을 확인할 수

있다. 압축률 변환에 대한 결과를 보면 98%의 높은 검출률을 보이지만 그에 반해 데이터베이스의 서로 다른 동영상의 2%를 질의 클립과 일치한다고 판단하여 출력하게 되므로 검색의 의미가 떨어질 수 밖에 없다. 그리고 Hua 방법은 Kim 방법보다는 높은 검출률과 긍정 오류율을 보이지만 제안한 단순 방법에 비해서는 낮은 성능을 보인다.

다음은 본 논문에서 제안하는 고속 검색 방법의 성능을 평가하였고, 그 실험 결과는 표 5와 같다.

표 4와 표 5의 모든 방법에 대한 평균 검출률, 평균 위치 오류율, 평균 긍정 오류율을 다시 그림 6과 같이 그래프로 비교해보면, 검출률에서는 고속 검색 방법이 단순 검색에 비해서는 낮고 다른 방법보다는 높은 성능을 보인다. 그리고 위치 오류율이나 긍정 오류율에서는 제안한 단순 방법과 고속 방법이 서로 근소한 차이를 나타내며, 비교 방법보다는 매우 낮은 발생률로 좋은 성능을 보여준다.

추출 및 정합 속도의 측정과 식별자 크기 비교는 비디오 검색의 정확성과 함께 필수적으로 평가되어야 하는 항목이다. 비교 알고리즘의 속도와 식별자 크기를 보여주고 제안한 방법의 단순 방법에서 고속 방법을 사용함으로써 인한 성능의 손해와 비교해서 수행 속도와 식별자 크기에서는 얼마나 이익이

표 4. 동영상 변형에 따른 성능 비교  
Table 4. Performance comparison for video modifications

	Hua 방법			Kim 방법			제안한 단순 방법		
	검출률	위치 오류율	긍정 오류율	검출률	위치 오류율	긍정 오류율	검출률	위치 오류율	긍정 오류율
밝기 변화 (L)	0.922	0.078	0.0041	0.971	0.029	0.0190	1.000	0.000	0.0004
밝기 변화 (M)	0.863	0.137	0.0035	0.961	0.039	0.0190	0.990	0.000	0.0004
밝기 변화 (H)	0.882	0.118	0.0045	0.971	0.020	0.0196	1.000	0.000	0.0004
대비 변화 (L)	0.912	0.088	0.0045	0.961	0.029	0.0202	1.000	0.000	0.0004
대비 변화 (M)	0.873	0.127	0.0036	0.961	0.039	0.0203	1.000	0.000	0.0004
프레임률 변환 (L)	0.931	0.069	0.0039	0.000	0.392	0.0239	1.000	0.000	0.0004
프레임률 변환 (M)	0.922	0.078	0.0039	0.000	0.304	0.0393	1.000	0.000	0.0004
흑백 변환	0.912	0.078	0.0038	0.971	0.020	0.0196	1.000	0.000	0.0004
해상도 축소 (L)	0.745	0.078	0.0039	0.804	0.069	0.1083	0.824	0.000	0.0004
해상도 축소 (M)	0.912	0.088	0.0039	0.980	0.020	0.0202	1.000	0.000	0.0004
압축률 변환 (L)	0.912	0.088	0.0039	0.980	0.020	0.0213	1.000	0.000	0.0004
압축률 변환 (M)	0.922	0.078	0.0039	0.980	0.020	0.0218	1.000	0.000	0.0004
자막/로고 삽입 (L)	0.480	0.225	0.0030	0.676	0.029	0.0183	0.961	0.000	0.0004
자막/로고 삽입 (M)	0.275	0.127	0.0013	0.490	0.059	0.0136	0.824	0.000	0.0002
테두리 잘림	0.294	0.186	0.0016	0.716	0.078	0.0218	0.833	0.000	0.0002
평균	0.784	0.110	0.0035	0.761	0.078	0.0271	0.962	0.000	0.0004



표 5. 고속 알고리즘의 성능  
Table 5. Performance of fast algorithm

	제안한 고속 방법		
	검출률	위치 오류율	긍정 오류율
밝기 변화 (L)	0.892	0.000	0.0008
밝기 변화 (M)	0.882	0.010	0.0004
밝기 변화 (H)	0.882	0.000	0.0006
대비 변화 (L)	0.892	0.000	0.0008
대비 변화 (M)	0.902	0.010	0.0005
프레임률 변환 (L)	0.902	0.020	0.0006
프레임률 변환 (M)	0.882	0.010	0.0006
흑백 변환	0.912	0.020	0.0005
해상도 축소 (L)	0.765	0.000	0.0010
해상도 축소 (M)	0.902	0.000	0.0005
압축률 변환 (L)	0.882	0.020	0.0005
압축률 변환 (M)	0.863	0.020	0.0006
자막/로고 삽입 (L)	0.814	0.020	0.0004
자막/로고 삽입 (M)	0.745	0.010	0.0003
테두리 잘림	0.843	0.020	0.0004
평균	0.864	0.010	0.0006

표 6. 식별자 추출과 정합 시간 비교 (ms)  
Table 6. Extraction and matching time of signature (ms)

	추출 시간	정합 시간	
		1단계 정합	2단계 정합
Hua 방법	167,743.156	68,547.761	3,300,451.937
Kim 방법	269,161.601	512,595.488	
제안한 단순 방법	208,898.542	18,249.059	
제안한 고속 방법	228,076.505	579.072	

시간을 측정한다. 모든 추출 및 정합 시간은 입력 영상 또는 파일을 읽는 과정부터 결과를 저장하는 모든 과정을 처리하는 시간을 측정하였다. Hua 방법의 2단계 정합에서 두 번째 정합 과정은 프레임의 삽입 및 제거 변형을 위한 과정으로 본 논문의 실험 환경에서는 그 필요성이 약하다. 그러므로 Hua 방법은 1단계 정합을 이용한 식별자 정합 시간을 함께 측정하였다. 각 알고리즘의 식별자 추출과 정합의 소요 시간은 표 6과 같다.

정합 속도의 실험 결과를 보면 제안한 알고리즘이 나머지 두 알고리즘에 비해 현저하게 빠른 수행 속도를 보여주고 있다. 제안한 알고리즘의 고속 방법을 사용하지 않더라도 정합 속도는 비교 알고리즘에 비해 몇 배 이상 빠르며, 고속 방법의 적용 시에는 성능 저하에 비해 매우 큰 속도 개선이 있음을 확인할 수 있다. 추출 속도에 대한 결과는 알고리즘의 복잡도는 물론이고 정규화 프레임률의 적용에 따라 영향을 받는다. 일반적으로 식별자 추출은 응용 방식에 따라 조건이 달라지겠지만, 대부분의 경우에는 오프라인에서 식별자를 추출하기 때문에 크게 평가 기준으로 두지 않는 경우가 많다.

다음으로 식별자 크기는 약 1분의 동영상 클립 10개에 대한 평균으로 측정하였다. 표 7의 실험 결과를 보면 제안한 알고리즘의 단순 방법이 식별자

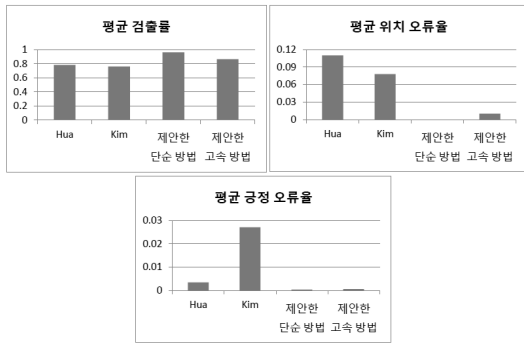


그림 6. 비교 방법과 제안 방법의 평균 성능 비교  
Fig. 6. Performance comparison between competition methods and proposed methods

되는지 확인할 수 있다.

식별자의 추출과 정합 속도를 측정하기 위해 10개의 원본 동영상을 이용하였고, 2개의 클립에 15가지 변형을 가해 전체 30개의 질의 클립을 만들어 이용하였다. 10개의 원본 동영상과 30개의 질의 클립의 총 길이는 83분 37초이다. 추출 속도는 전체 원본 동영상과 질의 클립에서 식별자의 추출 시간을 측정하고, 정합 속도는 30개의 질의 클립의 식별자를 10개의 원본 동영상의 식별자와 정합하는

표 7. 식별자 크기 비교 (bytes)  
Table 7. Signature size (bytes)

	Hua	Kim	제안한 단순 방법	제안한 고속 방법
추출 프레임률	10fps	29.97fps	4fps	4fps
식별자 크기	5,413	7,204	8,680	396

크기가 가장 크고 그 다음으로는 Kim 방법, Hua 방법 순으로 식별자 크기가 크다. 반면 제안한 고속 방법의 경우에는 다른 방법에 비해 열 배 이상의 매우 작은 크기만을 사용한다.

#### IV. 결 론

본 논문에서는 공간적 특징과 시간적 특징을 사용하여 식별자를 구성하고 고속 정합을 수행하는 동영상 고속 검색 방법을 제안하였다. 공간적 특징은 동영상 정합의 정확도를 높이기 위해 휘도 분포를 이용하여 정밀한 특징을 추출하고, 시간적 특징은 앞에서 추출한 공간적 특징으로 동영상의 시간적 유사성 지도를 생성하여 컷 검출을 통해 추출하였다. 제안한 방법에서 공간적 특징만을 이용한 결과를 보면 성능 면에서 96.2%의 만족할만한 결과가 나왔지만, 이 방법을 대용량 데이터베이스에 적용하기에는 추출되는 식별자의 크기와 검색 속도에서 문제가 된다. 이러한 문제를 해결하기 위해 시간적 특징을 추출하여 시간적 특징과 키프레임 공간적 특징열을 식별자로 구성하고 시간적 특징과 공간적 특징의 단계적인 정합 방법으로 고속 검색을 수행하였다. 고속 검색 방법의 실험 결과에서 86.4%의 성능을 확보하면서 식별자의 크기에서 10배 이상, 속도 면에서 30배 이상의 개선을 보였다. 그리고 고속 검색 방법의 성능은 비디오 검색의 대표적인 비교 기술보다도 정확성, 식별자 크기, 검색 속도 모두에서 높은 성능을 보여주었다. 이와 같이 성능 평가 실험을 통해서 제안한 알고리즘이 대용량 데이터베이스에서 효율적이고 효과적인 동영상 검색을 수행할 수 있음을 확인하였다.

그러나 제안한 고속 검색 방법은 동영상에서 컷 발생이 적은 경우에 식별자 내의 정보가 부족하기 때문에, 이런 경우는 동영상 검색 성능이 떨어지게 된다. 즉, 제안하는 고속 검색 방법은 컷 검출에 의해 키프레임의 공간적 특징만을 식별자의 요소로 구성하고 있기 때문에, 두 동영상의 부합을 판단할 만큼 최소한의 식별 정보가 보장돼야 한다. 그러므로 컷 정보가 부족한 동영상에 대해서도 정확한 검색이 가능하도록 알고리즘을 보완하여 그 성능을 개선시킬 수 있는 연구가 필요하다.

#### 참 고 문 헌

[1] J. Law-To, L. Chen, A. Joly, I. Laptev, O.

Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video Copy Detection: a Comparative Study," *Conference on Image and Video Retrieval*, 2007.

[2] X.-S. Hua, X. Chen, and H.-J. Zhang, "Robust Video Signature Based on Ordinal Measure", *International Conference on Image Processing*, 2004.

[3] C. Kim and B. Vasudev, "Spatiotemporal Sequence Matching for Efficient Video Copy Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 1(15):127-132, Jan., 2005.

[4] L. Chen, F.W.M. Stentiford, "Video sequence matching based on temporal ordinal measurement," *Pattern Recognition Letter*, 29(13), 1824-1831, Oct., 2008.

[5] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Trans. Multimedia*, Vol.9, No.2, Feb., 2007.

[6] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa, "Robust voting algorithm based on labels of behavior for video copy detection," *ACM Multimedia*, 2006.

[7] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Video," *Proceedings of IEEE ICCV*, 1470-1477, Oct., 2003.

[8] C.-Y. Chiu, C.-C. Yang, and C.-S. Chen, "Efficient and Effective Video Copy Detection Based on Spatiotemporal Analysis," *Proceedings of IEEE Symposium on Multimedia*, 2007.

[9] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol.60, No.2, pp.91-110, Nov., 2004.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," *Journal of Computer Vision and Image Understanding*, 110, 3, 346-359, 2008.

[11] I. H. Cho, A. Y. Cho, J. W. Lee, J. K. Jin, W. K. Yang, W. G. Oh, and D. S. Jeong, "Very Fast Concentric Circle Partition-Based Replica Detection Method", LNCS, *Advances in Image*

and Video Technology, Vol.4872, 2007, pp.905-918.

[12] www.open-video.org

[13] D. P. Heyman, and T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic," *IEEE/ACM Trans. Networking*, Vol.4, No.1, Feb., 1996.

임 예 은 (Ye-Eun Lim)

준회원



2009년 2월 인하대학교 전자공학과 학사

2009년 3월~현재 인하대학교 전자공학과 석사과정

<관심분야> 영상처리, 물체인식, 물체추적, 내용기반 검색

조 아 영 (Ayoung Cho)

정회원



2003년 2월 인하대학교 전자공학과 학사

2005년 2월 인하대학교 전자공학과 석사

2005년 3월~현재 인하대학교 전자공학과 박사과정

<관심분야> 영상처리, 영상 및 동영상 식별자, 워터마킹, 객체 검출

정 동 석 (Dong-Seok Jeong)

종신회원



1977년 2월 서울대학교 전기공학과 학사

1985년 Virginia Tech 전자공학과 공학석사

1988년 Virginia Tech 전자공학과 공학박사

1988년~현재 인하대학교 전자공학과 교수

1998년~2004년 정보전자공동연구소 소장

2010년~현재 인하대학교 공과대학 학장

<관심분야> 영상 및 동영상 처리, 영상 및 비디오 식별자

양 원 근 (Won-Keun Yang)

정회원



2004년 2월 인하대학교 전자공학과 학사

2006년 2월 인하대학교 전자공학과 석사

2006년 3월~현재 인하대학교 전자공학과 박사과정

<관심분야> 영상 및 동영상 식별자, 영상처리, 영상 검색

조 주 희 (Ju-Hee Cho)

준회원



2009년 2월 인하대학교 전자공학과 학사

2009년 3월~현재 인하대학교 전자공학과 석사과정

<관심분야> 영상처리, 컴퓨터 비전, 패턴인식, 내용기반 검색