

논문 2009-46SP-3-2

H.264/AVC로 압축된 비디오로부터 시그니처 추출방법

(Signature Extraction Method from H.264 Compressed Video)

김성민*, 권용광*, 원치선*

(Sung-Min Kim, Yong-Kwang Kwon, and Chee-Sun Won)

요약

본 논문은 H.264/AVC 비디오 압축영역에서 비디오 복제 방지기법의 일종인 CBCD(Content Based Copy Detection)에 사용될 수 있는 비디오 시그니처(Video Signature) 추출 방법을 제안한다. 기존의 비디오 시그니처 추출방법은 모두 비디오 공간영역에서 수행되기 때문에 압축된 비디오 스트림으로부터 시그니처를 추출하기 위해서는 비디오를 모두 복호해야 하는 단점을 가지고 있었다. 하지만 제안하는 방법에서는 비디오 압축영역에서 썸네일(Thumbnail)을 빠르게 구성하고 구성된 썸네일을 이용하여 비디오 시그니처를 추출하여 이와 같은 단점을 극복하였다. 밝기 순서 정보를 추출하는 실험결과로부터 제안하는 방법은 기존의 방법보다 80.98%의 정확도를 유지하면서 약 2.8배 빠르게 시그니처를 추출할 수 있었다.

Abstract

This paper proposes a compressed domain signature extraction method which can be used for CBCD (Content Based Copy Detection). Since existing signature extraction methods for the CBCD are executed in spatial domain, they need additional computations to decode the compressed video before the signature extraction. To avoid this overhead, we generate a thumbnail image directly from the compressed video without full decoding. Then we can extract the video signature from the thumbnail image. Experimental results of extracting brightness ordering information as the signature for CBCD show that our proposed method is 2.8 times faster than the spatial domain method while maintaining 80.98% accuracy.

Keywords: H.264/AVC, Video Signature, Video Identifier, Content Based Copy Detection

I. 서론

UCC(User Created Contents)는 최근 가장 대두되는 인터넷 트렌드(internet trend)이다. 특히, 동영상 UCC는 보다 적극적으로 자신을 표현할 수 있는 방법으로써 인터넷의 새로운 응용 모델이 되고 있으며 수많은 UCC 동영상들이 제작되어 유통되고 있다. 이에 따라, 네이버(NAVER)^[1], 다음(Daum)^[2] 등과 같은 포털 사이트들은 UCC 검색기능을 추가하고 UCC관련 채널을 늘

리고 있으며, 유튜브(YouTube)^[3]와 같은 UCC 전문 사이트들도 높은 방문자 수를 기록하고 있다. 하지만, 비영리적인 목적으로 만들어지는 초창기의 UCC와는 달리 최근에는 기존의 상업적 콘텐츠를 가공 변형하여 유통시킴으로써 저작권 침해에 대한 지적도 많아지고 있으며, 미국의 비아컴(viacom)은 구글(google) 소유의 유튜브에게 10만여 개 동영상의 삭제 요구와 동시에 10억 달러의 손해배상을 청구한 사실은 이러한 문제점의 심각성을 보여준다^[4].

이와 같은 무분별한 저작권 침해를 막기 위하여 워터마킹(watermarking), 콘텐츠 기반의 복제 검출(CBCD: Content-Based Copy Detection)과 같은 동영상의 복제 방지를 위한 노력은 그 동안 많이 진행되어 왔다^[5-6]. 워터마킹의 경우는 저작권 보호코드를 원본 동영상에

* 학생회원, ** 정회원, 동국대학교 전자공학과
(Dept. of Electronics Engineering, Dongguk University)

※ 이 논문은 2008년도 동국대학교 연구년 지원에 의하여 이루어졌음

접수일자: 2008년8월27일, 수정완료일: 2009년4월6일

눈에 보이지 않는 방법으로 삽입하는 기술이며, CBCD는 동영상의 특징을 일종의 메타데이터(metadata) 형태로 저장하여 이를 기반으로 복제된 영상을 검색하여 찾아내는 기술이다. 하지만 워터마킹 기술은 비디오의 저장형태 및 워터마크가 삽입된 비디오의 복제 시 적용되는 다양한 편집에 따라 워터마크 검출이 어려울 뿐만 아니라, 워터마크 삽입 후 동영상의 비트율 및 화질에 큰 영향을 끼치는 단점을 가지고 있다. 이와 같은 문제는 CBCD와 같이 동영상의 비트율 및 화질에 전혀 영향을 주지 않는 동영상 저작권 보호기법을 이용하여 해결 될 수 있기 때문에 최근의 연구는 CBCD를 중심으로 발전하고 있으며, ISO(International Standard Organization) 산하의 MPEG(Moving Picture Experts Group)-7 Visual Group에서도 최근 '비디오 시그니처(Video Signature)'라는 이름으로 CBCD와 같은 주제를 가지고 표준화에 박차를 가하고 있다^[7]. 하지만, 지금까지의 모든 CBCD 기법들은 동영상의 공간영역에서 수행되어 왔다. 따라서 모든 알고리즘은 압축되어 있는 동영상에 대해 복호화를 거친 후 시그니처를 추출하여 데이터베이스(Database)에 저장되어 있는 원본 시그니처와의 비교를 통하여 영상의 복제 유무를 판단하게 된다. 이러한 방법은 명백하게 많은 계산량을 필요로 하며, 특히 최근 방송 표준으로 정해진 H.264/AVC와 같은 복잡한 코덱이 동영상 압축을 위해 적용되었을 때 계산량은 더욱 더 많이 증가할 것이다. 이와 같은 문제를 해결하기 위하여, 본 논문에서는 효율적인 복제검출을 위하여 H.264/AVC로 압축된 비디오로부터 직접 시그니처를 추출하는 방법을 제안한다.

본 논문에서는 특히 비디오의 CBCD에 잘 적용되는 것으로 알려진 영상 내 영역 단위의 밝기 순서 정보를 H.264/AVC 압축영역에서 부분적인 디코딩을 통해 추출하는 알고리즘을 제안한다.

본 논문은 다음과 같이 구성되어 있다. II장에서는 이전의 연구결과와 이를 이용한 시그니처 추출 방법을 자세히 나타내며, III장에서는 실험결과를 그리고 IV장에서는 결론을 맺는다.

II. 본 론

그림 1에 일반적인 비디오 시그니처를 이용한 CBCD 방법을 나타내었다. 실제 모든 CBCD들은 이와 같이 비디오 공간영역에서 수행된다^[8~11]. [8]에서는 원본 비디오

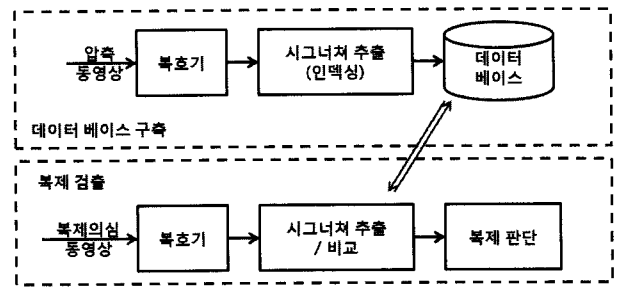


그림 1. 일반적인 비디오 시그니처를 이용한 콘텐츠 기반 복제 검출 방법의 블록도

Fig. 1. Block diagram of typical CBCD method using video signature.

오에 어떠한 편집도 가하지 않은 상태에서 비디오 복제 시 동영상 압축 코덱 종류 및 파라미터에 따라 콘텐츠의 색 정보가 바뀌어 복제 검출성능이 저하되는 문제를 지적 하였다. 이를 해결하기 위하여 비디오 각 프레임 사이의 움직임 정보, 각 화면내의 밝기 정보 및 일정 구간의 색깔 히스토그램(histogram)을 이용하여 비디오의 복제 유무를 판단하였다. 하지만, 이와 같은 방법은 현재 비디오 복제 시 빈번하게 행해지고 있는 여러 편집 기술에 대해서는 복제 검출 결과를 보장하지는 못한다.

이를 해결하기 위하여 [9]에서는 비디오 복제 시 가장 흔하게 발생할 수 있는 화면 밝기 변화, 히스토그램 균일화(histogram equalization), 레터-박스 스타일(letter-box style)로의 변환, 4:3에서 16:9로의 종횡비 변환 등에 강인한 검출 방법을 위하여 동영상 각 프레임을 일정 구간으로 나누고 각 구간사이의 평균 휘도 순서정보를 이용하는 CBCD방법을 제안하였다. 실험 결과로부터 [9]에서 제안하는 방법은 다른 방법들과는 달리 화면 내 비디오의 밝기 정보와 밝기 정보의 시간 축 변화 정보만을 이용하여 매우 우수한 복제 검출 결과를 나타내었다. 이와 같은 방법은 비 전문가에 의한 단순한 복제 검출에는 강인할 수 있지만, 콘텐츠의 배경을 완전히 바꾸는 등의 점점 다양화되는 비디오 편집 기술에 의한 복제 검출에는 알맞지 않은 단점이 있다.

이러한 의도적인 편집을 추가하는 비디오 복제의 경우 [10~11]과 같은 방법으로 복제 유무를 판단 할 수 있다. [10~11]에서는 [12]에 나타난 Harris와 Stephens의 코너(corner) 정보를 시그니처로 사용하였다. 즉, [10]의 논문에서는 영상의 에지정보는 영상에 가해진 여러 편집 종류에 따라 크게 바뀔 수도 있고 약간의 회전공격에도 방향 성분이 크게 변할 수 있지만, 영상이 가지고 있는 중요 피사체의 코너 정보는 영상에 대한

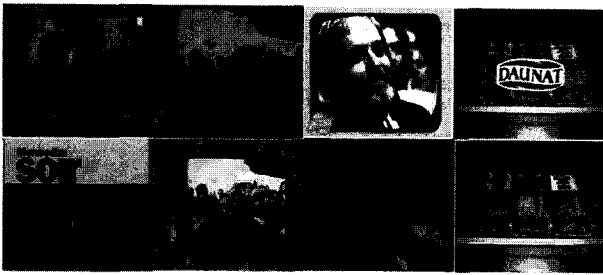


그림 2. 다양한 편집을 포함하는 비디오 복제의 예
Fig. 2. Examples of video copy with various editing

어떠한 편집에도 비교적 강인하다는 점을 이용하였다. 하지만 [10]의 방법은 비디오 프레임으로부터 추출한 방대한 양의 비디오 시그너처 정보의 저장공간 및 많은 검색시간을 필요로 하는 문제를 가지고 있다. 이러한 문제를 보완하기 위해 [11]에서는 장면전환 검출 후 [12]의 방법을 이용하여 시그너처를 추출하였다. 이와 같은 방법은 그림 2와 같은 비디오 편집을 포함하는 비디오 복제공격에도 매우 강인하게 나타나는 것으로 알려져 있다.

이러한 모든 방법은 비디오의 공간영역에서 수행된다. 그러나 대부분의 비디오의 유통은 압축된 상태에서 수행되기 때문에 CBCD를 이용하기 위해서는 우선 압축 비디오를 복호하여 공간영역으로 바꾼 후 비디오 복제 검출에 이용하여야 한다. 이는 명백하게 많은 계산을 필요로 하며, 특히 최근 비디오 압축 표준으로 채택된 H.264/AVC의 경우 이전의 압축 표준이었던 MPEG-2 보다 더 많은 계산량을 필요로 하기 때문에 [13] H.264/AVC로 압축된 비디오에 대한 기존의 CBCD 방법들 역시 계산상의 문제가 더욱 심각해진다.

본 논문에서는 이와 같은 방법을 해결하기 위하여 H.264/AVC 압축영역에서 비디오 시그너처를 생성하는 방법을 제안하며, 비디오 압축영역에서 시그너처를 추출하여 빠른 CBCD 시스템의 핵심 기술로 이용될 수 있다. 즉, 그림 3에 나타낸 바와 같이 제안하는 시스템은 기존의 CBCD를 위해 반드시 사용해야 하는 비디오 복호기를 사용하지 않고, 비디오 압축영역에서 시그너처를 추출하게 된다. 이를 위하여 이전 연구[14][15]에서처럼 H.264/AVC 인트라 프레임(intra frame)의 압축영역으로부터 추출한 DC계수와 인트라 예측 모드(intra prediction mode)로부터 섬네일 영상을 만들고, 만들어진 섬네일 영상으로부터 [9]에서 사용된 시그너처를 추출하는 방법을 적용하였다.

1. H.264/AVC 압축영역에서 섬네일 추출방법

그림 4에 H.264/AVC 압축영역에서의 섬네일 추출방법을 그림으로 나타내었다.

제안된 섬네일 추출방법은 기존의 H.264/AVC 디코더와 비교하였을 때 [16], 역 변환(inverse transform) 부분과 움직임 보상(motion compensation)이 제거되었고 단순히 인트라 프레임의 DC값과 예측 모드(intra prediction mode)만이 비트스트림에서 추출되어 최소한의 연산으로 빠르게 섬네일 영상을 구성한다. 하지만 H.264/AVC의 인트라 예측의 영향으로 DC계수만 이용하여 섬네일 영상을 만들었을 시에는 심한 에러전파가 야기되며, 이를 해결하기 위해서는 그림 5에 나타낸 바와 같이 1번과 2번 픽셀을 정확하게 복호하는 것이 가장 중요하다 [15]. 이때 실제 섬네일 영상에는 4x4 블록 단위의 가장 마지막 픽셀이 사용된다. 즉, 그림 5의 각 블록에서 중요도가 높은 1번과 2번 픽셀 모두를 복호되 나머지 픽셀의 복호는 생략해도 약 절반의 계산량으로 섬네일 영상을 생성할 수 있다. 단, 16x16 휘도성분 예측모드와 8x8 색차 성분 예측모드의 3번 픽셀은 이웃 블록의 복호과정에는 영향을 주지 않으므로 계산량과 구조적 복잡성을 줄이기 위해 해당 블록에서 복호된 픽셀의 대표 값을 단순 대입한다.

그림 5의 NxN블록에서 픽셀 좌표의 행과 열을 0에

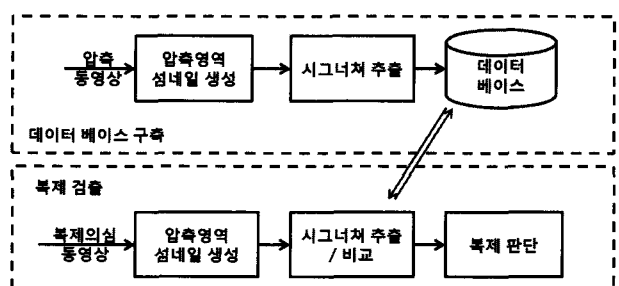


그림 3. 제안하는 시스템의 블록도
Fig. 3. Block diagram of the proposed system.

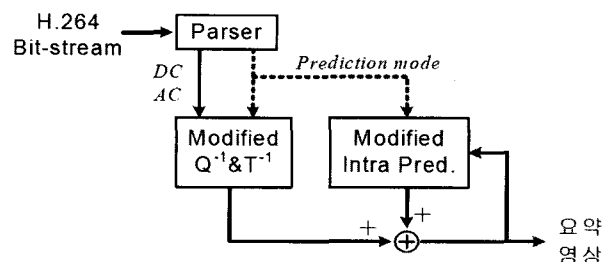


그림 4. H.264/AVC 압축영역에서의 섬네일 추출방법
Fig. 4. Thumbnail image extraction method in H.264/AVC Compression Domain

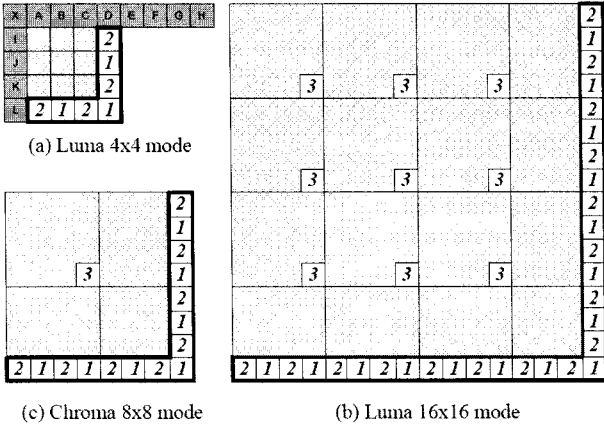


그림 5. 제안하는 방법에서 복호되는 픽셀
Fig. 5. Decoded pixel by proposed method.

서 (N-1)의 값을 갖는 i, j 로 각각 정의하면, 한 블록에
서 요약영상을 위한 복호된 픽셀 X'_{ij} 를 구하는 과정은
다음 수식 (1)과 같이 나타낼 수 있다.

$$X'_{i,j} = \begin{cases} \frac{(T_{i,j} + (P_{i,j} \times 2^6) + 32)}{2^6} & i, j = N - 1 \\ skip & otherwise \end{cases} \quad (1)$$

$T_{i,j}$ 와 $P_{i,j}$ 는 각각 블록의 (i, j)위치에서의 역양자화
및 역변환의 결과값과 예측모드에 의해 계산된 값이며,
 $j=N-1$ 혹은 $i=N-1$ 인 경우에만 연산이 이루어진다^[12].
식에서 2^6 이 사용된 이유는 H.264 표준에서 연산 속도
증가와 실수 연산에서 발생하는 오차를 배제시키기 위
해 DCT에서 정수형 연산을 수행하기 때문이다. 즉, 정
수형 DCT 연산을 위해 잔여 데이터에 64를 곱하여 소
수 이하의 값을 일부 정수연산에서 사용하게 한다. 따
라서 복호 시에는 $P_{i,j}$ 에 64를 곱하여 추가적인 반올림
문제의 해결을 위한 값 32와 함께 $T_{i,j}$ 에 더하고 64로
나누어야 한다.

추가적인 연산량 감소를 위해 그림 6과 같이 특정 위
치의 픽셀만을 복호하는 방법도 고려될 수 있다. 예를
들어 블록의 예측모드가 Horizontal-Down (27.5°)인 경
우, 일반적인 자연영상에서는 이웃 픽셀간의 상관성이
높기 때문에 그림 6에서와 같이 X, B, D, F, H, J 그리
고 L만을 참조픽셀로 사용하여 예측블록을 추정할 수
있다. 즉, A, I, K는 이웃 참조픽셀 B, J, L 그리고 X를
이용한 평균값으로 사용하여 예측값 $P_{i,j}$ 를 추정할 수
있으며, 식 (2)와 같이 정리된다.

필요한 참조픽셀

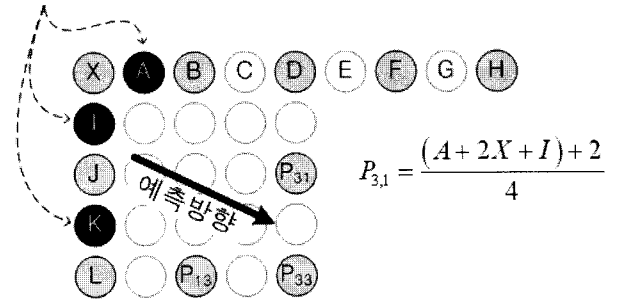


그림 6. 제안하는 방법에서의 4x4 블록 복호 방법
Fig. 6. The decoding method of 4x4 block.

$$P_{3,1} = (A + 2X + I + 2)/4 \quad (2.a)$$

$$P_{3,3} = (I + 2J + K + 2)/4 \quad (2.b)$$

$$P_{1,3} = (J + 2K + L + 2)/4 \quad (2.c)$$

그 결과를 이용하여 그림 6과 같이 서브블록을 생성
한다. 이렇게 생성된 서브 블록은 계산량을 줄이기 위
하여 모든 픽셀을 복호하는 대신 $P_{i,j}$ 만을 이용하여 영
상을 구성하기 때문에 영상의 화질은 원본과 차이가 있
을 수 있지만, 제안하는 CBCD방법에 이용하는 데에는
큰 영향을 주지 않기 때문에 빠른 CBCD 시스템을 구
성하기에 알맞다.

2. 시그너처 생성방법

본 연구에서는 이와 같은 H.264/AVC 압축영역으로부터
서브네임을 추출하고, 추출된 서브네임 영상으로부터 시
그너처를 추출하였다. 사용된 시그너처 추출 방법은 [9]
의 방법과 비슷하다. 그림 7에 [9]에서 사용하는 시그너
처 추출 방법을 나타내었다. 만약 n 개의 프레임을 가지
는 비디오의 각 프레임을 m 개의 블록으로 나눈다면 각

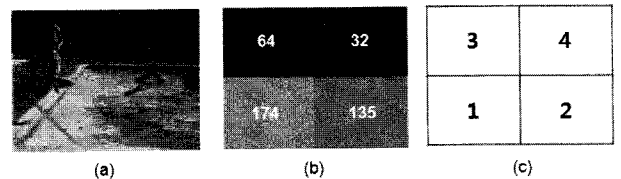


그림 7. [9]에서 사용하는 비디오 시그너처 생성방법.
(a)원본영상 (b)2x2로 영상을 나눈 후 각 블록의
평균 휘도를 계산 (c)순위 행렬 작성
Fig. 7. Video Signature Extraction Method of proposed
by [9]. (a) Original image (b) Average values for
partitioned 2x2 blocks (c) Rank matrix of (b).

프레임의 시그너처는 $V^m[n]$ 으로 나타낼 수 있다.

그림 7에는 [9]의 실험과 일치시켜 $m=4$ 인 경우를 나타내었다. 그림 7-(a)과 같이 비디오 프레임을 4개의 블록으로 나눈다면, 각 블록에 대한 평균휘도는 그림 7-(b)와 같이 나타낼 수 있으며, 각 휘도 수치를 내림차순으로 정리하면 그림 7-(c)와 같이 나타낼 수 있다. 따라서 프레임의 좌측 상단으로부터 우측 하단으로 순서대로 시그너처를 생성하면, $V^0=3$, $V^1=4$, $V^2=1$, 그리고, $V^3=2$ 이다. 이와 같은 시그너처 추출은 비디오 전체 프레임에 대하여 수행된다.

III. 실험

실험에서는 다양한 표준영상 비디오를 H.264/AVC 인트라 부호화 하여, 제안하는 섬네일 영상 추출기법으로 시그너처를 추출하였다. 또한, 기존 연구[9]에서 추출된 시그너처와 비교하여 제안하는 H.264/AVC 압축영역에서 추출한 시그너처의 사용가능 여부를 확인



그림 8. (a) 원본영상과 (b) 추출한 섬네일 영상
Fig. 8. (a)original image and (b)extracted thumb-nail image.

하였다. 그림 8에는 원본영상과 추출한 섬네일 영상을 나타내었다.

그림 8에 나타낸 바와 같이 본 논문에서 제안하는 H.264/AVC 압축영역에서 추출한 섬네일 영상은 H.264/AVC 압축에서 부호화의 기본단위인 4x4블록에서 대표값이 추출되어 1/16 크기의 영상이 섬네일로 생성된다. 또한 복호화 단계에 적용되는 인트라 예측 정보를 간략화 하였기 때문에 원본영상과 픽셀수치가 조금 다르게 나타나지만, 영상이 가지는 내용의 변화는 전혀 없다.

표 1, 2에는 기존 방법과의 성능비교를 나타내었다. 표에 나타난 바와 같이 실험에서는 16개의 표준영상 약 7000프레임으로부터 시그너처를 생성하였다. 이때 다양한 비교를 위하여 표 1은 $m=9$ (즉, 3x3)일 때의 [9]의 방법과 제안방법의 성능비교를, 표 2는 [9]의 실험과 일치하는 $m=4$ (즉, 2x2)일 때의 [9]의 방법과 제안방법의 성능비교를 각각 나타낸다. 각 표준영상은 500Kbps의 비트율로 인트라 부호화 하였다.

표 1에 나타낸 바와 같이 제안하는 시그너처 추출방법은 기존 [9]의 방법보다 $m=9$ 와 4일 때 각각 2.81배 와 2.83배로 빠르게 시그너처를 추출하고 있다. 이때, 시그너처 추출과정은 블록의 평균 휘도 값을 측정하기 때문에 대부분의 연산은 각 픽셀휘도 수치를 더하는 연산이다. 따라서 표 1, 2에 나타난 바와 같이 제안하는 방법의 시그너처 추출에 소요되는 시간은 블록의 개수 m 에 상관없이 거의 비슷함을 알 수 있다.

또한, 본 논문에서는 비디오 압축영역에서 추출된 시그너처의 공간영역에서 추출된 시그너처에 대한 정확도를 알아보기 위하여 다음 수식 (3)과 같은 L-1 Norm을 사용하였다.

$$D = \frac{1}{mn} \sum_{f=0}^n \sum_{p=0}^m |V_s^p[f] - V_c^p[f]| \quad (3)$$

이때, $V_s^m[n]$ 는 비디오 공간영역에서 n 개의 프레임 을 갖는 비디오의 각 프레임을 m 개의 서블 블록으로 나누어 추출한 시그너처를, $V_c^m[n]$ 는 해당 비디오의 압축영역에서 추출한 시그너처를 나타낸다. D 의 최소값은 각 블록의 순서정보가 정확히 일치하는 경우 $m=9$ 와 4일 때 모두 0의 수치로 나타나며, 최대값은 블록의 순서정보가 반대로 형성되는 경우 $m=9$ 와 4에서 각각 4.44 와 2의 수치로 나타난다.

표 1. 성능 비교 (m=9)

Table 1. Performance Comparing (m=9).

표준영상	프레임수	[9]의 방법 (sec)	제안 방법 (sec)	속도 향상 (배)	정확도 (D)
bridge	2101	134.64	45.82	2.94	0.617
bus	150	9.34	3.43	2.73	0.513
coastguard	300	19.56	7.44	2.63	0.428
container	250	19.44	6.54	2.97	0.320
flower	300	15.39	5.85	2.63	0.573
foreman	300	19.25	5.95	3.24	0.561
hall	300	19.17	6.54	2.93	0.626
highway	2000	128.56	43.31	2.97	0.614
mobile	300	18.64	7.54	2.47	0.424
news	300	19.47	6.85	2.84	0.391
silent	300	18.84	6.65	2.83	0.542
stefan	90	5.44	2.02	2.69	0.555
tempete	260	15.86	5.73	2.77	0.579
waterfall	260	16.00	5.83	2.75	0.421

표 2. 성능 비교 (m=4)

Table 2. Performance Comparing (m=4).

표준영상	프레임수	[9]의 방법 (sec)	제안 방법 (sec)	속도 향상 (배)	정확도 (D)
bridge	2101	134.64	45.76	2.94	0.203
bus	150	9.34	3.41	2.74	0.181
coastguard	300	19.56	7.42	2.64	0.17
container	250	15.39	6.52	2.36	0.147
flower	300	19.44	5.81	3.35	0.201
foreman	300	19.25	5.92	3.25	0.191
hall	300	19.17	6.51	2.94	0.214
highway	2000	128.56	43.26	2.97	0.211
mobile	300	18.64	7.52	2.48	0.166
news	300	19.47	6.82	2.86	0.151
silent	300	18.84	6.62	2.85	0.194
stefan	90	5.44	2.00	2.71	0.184
tempete	260	15.86	5.71	2.78	0.187
waterfall	260	16.00	5.81	2.75	0.154

표 1, 2에 나타난 바와 같이 제안하는 압축영역의 시그니처 추출 방법은 공간영역에서 추출한 시그니처와 비교할 때 평균적으로 m=9와 4에서 각각 D=0.511와 0.19로써 기존의 방법과 큰 차이가 없음을 알 수 있다.

표 3, 4에는 프레임당 H.264/AVC 압축영역으로부터 추출한 순서정보가 공간영역에서 추출된 각 프레임의

표 3. 공간영역 순서정보와 압축영역 순서정보의 차이 비교 표 (% , m=9)

Table 3. Accuracy comparison between Spatial Domain Ordering and Compression Domain Ordering (% , m=9).

$V_s \setminus V_c$	1	2	3	4	5	6	7	8	9
1	76	10.7	8.3	4	0.3	0.3	0.3	0	0
2	13.7	46.7	36.3	3.3	0	0	0	0	0
3	9.7	38.7	40.3	10	1.3	0	0	0	0
4	0.7	4	13.3	60	20	1	0.7	0.3	0
5	0	0	1.7	17.3	60.3	15	4.3	1	0.3
6	0	0	0	4.3	10.7	67	8.3	5.3	4.3
7	0	0	0	0.7	4.3	8.7	51.7	20	14.7
8	0	0	0	0.3	2.7	3.3	21.3	52.7	19.7
9	0	0	0	0	0.3	4.7	13.3	20.7	61

표 4. 공간영역 순서정보와 압축영역 순서정보의 차이 비교 표 (% , m=4)

Table 4. Accuracy comparison between Spatial Domain Ordering Information and Compression Domain Ordering (% , m=4).

$V_s \setminus V_c$	1	2	3	4
1	73.7	26	0.3	0
2	26.3	66	7.7	0
3	0	8	88	4
4	0	0	4	96

순서정보와 얼마큼 다른지를 표로 나타내었다. 표 3에 나타난 실험에서는 m=9(3x3)로 설정하였기 때문에 $1 \leq V_s, V_c \leq 9$ 이고, 표 4에 나타난 [9]의 방법과 일치하는 실험에서는 m=4(2x2)로 설정하였기 때문에, $1 \leq V_s, V_c \leq 4$ 다. 즉, 표 3, 4는 공간영역에서 각 프레임의 순서정보인 V_s 가 압축영역에서 추출한 V_c 로 변환된 정도를 백분율로 보여준다. 표 3의 음영부분을 예로 들면, 각 프레임의 서브 블록 중 두 번째로 휘도가 큰 블록($V_s = 2$)이 압축영역에서 추출 되었을 때(V_c) 전체 동영상에 대하여 평균적으로 첫 번째 휘도가 큰 블록으로 나타나는 경우가 13.7%, 두 번째 휘도가 큰 블록으로 나타나는 경우가 46.7%, 세 번째 휘도가 큰 블록으로 나타나는 경우가 36.3%, 네 번째 휘도가 큰 블록으로 나타나는 경우는 3.3% 임을 나타낸다.

이와 같은 정확도는 공간영역에서의 순서정보가 압

표 5. 순서정보의 통계적 수치 (%)
Table 5. Statistics for Ordering Information (%)

$ V_s - V_e $ m	0	1	2	3	4	5	6	7	8
m=9	57.36	31.63	8.76	2.09	0.17	0.03	0.03	0	0
m=4	80.98	19.02	0	0					

축영역에서 추출된 순서정보와 정확히 일치됨을 의미하며, 표 5와 같이 간략히 나타낼 수 있다.

표 5의 가로축은 V_s 와 V_e 의 차이값을 나타내며, 만약 0일 때는 공간영역의 순서정보가 압축영역의 순서정보와 일치하는 경우를 나타낸다. 표 5에 나타낸 바와 같이 공간영역의 순서정보는 압축영역의 순서정보와 정확히 일치하는 경우는 m=9, m=4일때 각각 57.36%, 80.98%이며, 해당 순서정보가 수치 1 차이 나는 경우는 각각 31.63%, 19.02% 이다. 즉, m=9 및 m=4일때 공간영역에서 추출한 각 프레임내 블록의 순서정보와 압축영역에서 추출한 각 프레임내 블록의 순서정보가 최대 1차이나는 경우는 각각 88.99%, 100% 이다. 특히 기존방법[9]과 일치하는 실험조건(m=4)에서는 최대 1차이 나는 경우가 100%이므로, 이는 제안하는 방법을 이용하여 추출한 시그니처가 충분히 사용가능하다는 것을 나타낸다.

IV. 결 론

본 논문에서는 기존의 비디오 공간영역에서 수행되던 시그니처 추출방법을 비디오 압축영역에서 직접 수행할 수 있는 방법을 제안하였다. 이를 위하여 H.264/AVC로 압축된 비디오를 직접 복호화하지 않고, 압축 영역에서 섬네일을 빠르게 추출하여, 추출된 섬네일로부터 시그니처를 추출하였다. 실험 결과로부터 제안하는 방법은 기존 [9]와 같은 실험조건에서 80.98%의 정확도를 유지하면서 약 2.8배 빠르게 비디오 시그니처를 추출할 수 있었다.

참 고 문 헌

[1] http://video.search.naver.com/search.naver?where=video&sm=tab_jum&query=UCC
 [2] <http://tvpot.daum.net/Top.do>
 [3] <http://kr.youtube.com/>
 [4] Paul R. La Monica, "Viacom sues 'GooTube' for \$1 billion," CNN News, Mar. 13 2007.

[5] Chuhong Fei, Deepa Kundur, and Raymond H. Kwong, "Analysis and Design of Secure Watermark-Based Authentication Systems," IEEE Trans. on Information Forensics and Security, vol. 1, no. 1, Mar. 2006.
 [6] Olivier Buisson, Valerie Gouet-Brunet, Nozha Boujemaa, and Fred Stentiford, "Video Copy Detection: a Comparative Study," ACM international Conference on Image and Video Retrieval, pp. 371-378, July 2007.
 [7] M. Bober, K. Iwamoto, and P. Brasnett, "Description of MPEG-7 Visual Core Experiment,"MPEG output document N9452, Oct. 2007.
 [8] Arun Hampapur, Ki Ho Hyun, and Ruud Bolle, "Comparison of Sequence Matching Techniques for Video Copy Detection," International conference on Storage and Retrieval for media Databases, 2002.
 [9] Changick Kim, and Bhaskaran Vasudev, "Spatiotemporal Sequence Matching for Efficient Video Copy Detection," IEEE Trans. on Circuits and Systems for Video Technology, vol. 15, no. 1, pp. 127-132, Jan. 2005.
 [10] Julien Law-to Olivier Buisson, "Robust Voting Algorithm Based on Labels of Behavior for Video Copy Detection," Proceedings of the 14th annual ACM international conference on Multimedia. pp.835-844, 2006.
 [11] Alexis Joly, Olivier Buisson, and Carl Frelicot, "Content-Based Copy Retrieval Using Distortion-Based Probabilistic Similarity Search," IEEE Trans. on Multimedia, vol. 9, no. 2, pp. 293-306, Feb. 2007.
 [12] C. Harris, and M. Stevens, "A Combined Corner And Edge Detector," 4th Alvey Vision Conference, pp. 153-158, 1988.
 [13] ATI white paper, "Introduction to H.264," http://ati.amd.com/products/pdf/h264_whitepaper.pdf
 [14] 권용광, 김성민, 원치선, "H.264의 인트라 예측모드와 DC를 이용한 섬네일 추출 방법," 한국통신학회 하계학술대회, vol. 35, July 2007.
 [15] 권용광, 원치선, "DMB를 위한 요약영상 추출 알고리즘," 한국방송공학회 논문지, 제 12권, 제 5호, 2007.
 [16] T. Wiegand, G. Sullivan, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans. on Circuit and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003.

저 자 소 개



김 성 민(학생회원)
 2000년 동국대학교 전자공학과
 학사
 2005년 동국대학교 전자공학과
 석사
 2005년 9월~현재 동국대학교
 전자공학과 박사과정
 <주관심분야 : 비디오 신호처리>



권 용 광(정회원)
 2002년 동국대학교 전자공학과
 석사
 2008년 동국대학교 전자공학과
 박사
 1999년~2002년 (주)METEL
 연구원
 2002년~현재 안양과학대학 강사
 <주관심분야 : H.264/AVC Codec, 영상처리>



원 치 선(정회원)-교신저자
 1982년 고려대학교 전자공학과
 학사
 1986년 University of
 Massachusetts at
 Amherst 석사
 1990년 University of
 Massachusetts at
 Amherst 박사
 1989년 11월~1992년 8월 금성사(현 LG전자)
 선임연구원
 1992년 9월~현재 동국대학교 전자공학과 교수
 <주관심분야 : 영상검색, 워터마크, 영상분할>