

논문 2006-43SP-5-9

# H.264/AVC 압축 영역에서의 동영상 검색

## ( Video Shot Retrieval in H.264/AVC compression domain )

변주완\*, 김성민\*, 원치선\*

(Byun Ju Wan, Kim Sung Min, and Won Chee Sun)

### 요약

본 논문은 H.264/AVC 압축 영역에서의 동영상 검색에 관한 것이다. 기존의 동영상 압축 표준인 MPEG-2 및 4와는 달리 H.264/AVC는 다양한 블록 단위의 움직임 보상으로 인해 기존의 MPEG-2 및 4에서 사용되었던 움직임 벡터를 이용한 방법들을 그대로 적용하기 어렵다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 H.264/AVC의 다양한 블록 크기의 움직임 벡터를 움직임 보상의 최소 단위인 4x4 블록 단위로 분할하고, 각각의 분할된 블록에 동일한 움직임 벡터를 할당하는 방법을 제안한다. 또한, 유사도 측정을 위해 두 영상으로부터 추출한 특징 간의 상관계수를 이용하였다. 실험은 10558 프레임의 MPEG 표준 동영상과 48161 프레임의 일반 동영상을 대상으로 이루어졌으며, 제안한 방법을 이용한 실험을 통해 0.2 이하의 ANMRR 결과를 나타내었다

### Abstract

In this paper, we present a video shot retrieval algorithm in H.264/AVC compression domain. Unlike previous standards such as MPEG-2 and 4, H.264/AVC supports a variable block size for motion compensation. Therefore, existing video retrieval algorithms exploiting the motion vectors in MPEG-2 and 4 domains are not appropriate for H.264/AVC. So, we devise a method to project motion vectors with larger than 4x4 block sizes into those for the smallest 4x4 blocks. It also uses correlations among features for the measure of similarity. Experimental results with standard videos of 10558 frames and commercial videos of 48161 frames show that the proposed method yields ANMRR less than 0.2.

**Keywords :** H.264/AVC, video shot retrieval, sequence matching.

### I. 서론

컴퓨터(computer)와 인터넷(internet)의 급속한 발전과 함께, 네트워크(network)를 통해 제공되는 동영상 정보의 요구 또한 크게 증가하였다. 그러나 만약 효과적인 동영상 분류 및 검색이 이루어지지 않는다면 사용자는 동영상 정보를 검색하기 위해 많은 시간을 소비하게 될 것이다. 이러한 문제를 해결하기 위해 효과적인

동영상 검색을 위한 많은 연구가 진행되어왔다<sup>[1-4]</sup>. 기존의 동영상 검색은 크게 공간 영역에서의 동영상 검색<sup>[1]</sup>과 MPEG(Moving Picture Experts Group)과 같은 동영상 부호화 표준을 이용한 압축 영역에서의 동영상 검색<sup>[2-4]</sup>으로 분류할 수 있다. 공간 영역에서의 동영상 검색은 다양한 영상 처리 기법을 통해 비교적 정확한 검색이 가능하다. 그러나 대부분의 동영상은 MPEG과 같은 동영상 부호화 표준에 의해 압축된 비트 스트림(bit-stream)의 형태로 저장 및 전송되고 있다. 따라서 공간 영역에서 동영상을 검색하기 위해서는 압축된 동영상 전체를 복호해야하는 단점이 있다. 이러한 단점을 극복하기 위해 압축된 비트 스트림 내에서 필요한 정보만을 추출해 동영상 검색에 이용하는 압축 영역에서의 내용 기반 동영상 검색 방법이 주로 사용되고 있다.

기존의 압축 영역에서의 동영상 검색은 주로 DCT

\* 정회원, 동국대학교 전자공학과  
(Dept. of Electronic Engineering, Dongguk University)

※ 이 논문은 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업(ITA-2005-(C1090-0502-0027)) 및 서울시 산학협력사업으로 구축된 서울 미래형콘텐츠 컨버전스 클러스터의 지원으로 수행되었음

접수일자: 2006년1월9일, 수정완료일: 2006년8월9일

(Discrete Cosine Transform) 계수를 이용한 방법<sup>[2]</sup>과 움직임 벡터(motion vector)를 이용한 방법<sup>[3, 4]</sup>이 사용되었다. 먼저, DCT 계수를 이용한 방법은 DC(Direct Current) 및 AC(Alternating Current) 계수로부터 추출한 영상의 색상, 밝기, 그리고 에지(edge)와 같은 영상의 다양한 특징 정보를 이용한다. 이러한 DCT 계수는 MPEG-2 비트 스트림 내에서 쉽게 추출할 수 있고 동영상에 가진 특성을 잘 표현할 수 있다. 그러나 기존의 MPEG-2 기반에서 사용되었던 DCT 계수를 이용한 방법들은 H.264/AVC의 인트라 예측(intra prediction)과 다양한 블록(block) 크기의 움직임 보상과 같은 새로운 동영상 부호화 기법<sup>[5-7]</sup>으로 인해 H.264/AVC에 그대로 적용되기 어렵다. 즉, DC 계수가 독립적으로 부호화되었던 MPEG-2 및 4와는 달리 H.264/AVC의 DC 계수는 AC 계수와 함께 부호화된다. 또한, H.264/AVC 비트 스트림으로부터 추출된 DC 계수는 인트라 프레임(intra frame)에 적용된 인트라 예측 기법으로 인해 현재 블록과 인접 픽셀(pixel) 간의 에너지(energy) 차를 나타낸다. 따라서, H.264/AVC의 DC 계수를 이용하기 위해서는 DC 계수의 역 예측 과정이 필요하고, 이 과정은 비트 스트림을 전체 복호화 하는 것과 비슷한 연산량을 가지므로 효과적이지 않다. 또한, H.264/AVC의 AC 계수는 공간 예측을 하지 않는 MPEG-2 및 4의 AC 계수보다 방향성에 대한 정보가 미약하기 때문에 한 프레임의 공간적 에지 성분을 나타내기에는 적절하지 않다. 그리고, 인터 프레임(inter frame)에서 일률적인 16x16 픽셀 크기의 움직임 보상을 하는 기존의 MPEG과는 달리 H.264/AVC는 다양한 크기(16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 픽셀 단위)의 움직임 보상 블록이 존재한다<sup>[5-7]</sup>. 즉, 기존의 MPEG과는 달리 H.264/AVC에서는 같은 값의 움직임 벡터일지라도 블록 크기에 따라 각각 다른 움직임 정도를 나타낸다. 이와 같이 H.264/AVC는 기존의 MPEG과는 다른 부호화 방식을 사용함에 따라 기존의 MPEG에서 사용되었던 동영상 검색 방법<sup>[2-4]</sup>을 H.264/AVC에 적용하는 것은 구현상의 복잡도, 적용 효과 등의 문제로 어려움이 따른다. 따라서 본 논문에서는 기존의 방법을 개선하여 H.264/AVC의 다양한 크기의 움직임 벡터를 움직임 보상의 최소 단위인 4x4 픽셀(pixel) 크기로 분할하고, 분할된 각각의 4x4 픽셀 블록에 기존의 움직임 벡터 값을 동일하게 할당하여 움직임 벡터를 재구성하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 H.264/AVC 비트 스트림으로부터 동영상 검색을 위한 특징

정보를 추출하는 과정을 설명한다. III장에서는 두 동영상, 즉, 질의 동영상 샷(shot)과 실험 동영상 간의 유사도를 측정하는 방법에 대해 설명한다. 그리고 IV장은 실험 결과와 검색 성능을 보이고, V장에서 본 논문의 결론을 맺는다.

## II. 동영상 검색을 위한 특징 추출

서론에서 언급한 바와 같이 멀티미디어 정보가 늘어남에 따라 동영상 정보를 검색하는 일도 많아지게 되었다. 하지만 기존의 문자 기반 정보 검색으로는 방대한 양의 멀티미디어 정보 속에서 사용자의 요구를 적절히 반영한 검색을 하기 어렵고, 따라서 인간 시각적인 관점에서의 내용 기반 동영상 검색이 요구된다<sup>[8]</sup>. 또한 동영상 검색은 정지 영상을 검색하는 것과는 달리 영상 내 인물, 사물 등의 내용뿐만 아니라 그 내용의 의미론적인 관점에서의 검색(semantic context video retrieval)이 필요하며, 검색 과정에서의 처리 속도 또한 중요시 되어야한다. 따라서 본 논문에서는 영상을 표현하는 많은 특징 정보들 가운데 연속된 영상 사이의 움직임 정도를 표현하는 움직임 벡터를 사용하였으며, 알고리즘의 복잡도를 가능한 최소화한 의미론적 내용 기반 동영상 검색 방법을 제안한다.

### 1. 움직임 벡터 추출

동영상을 효과적으로 압축하기 위해서는 공간적인 중복성과 시간적인 중복성, 그리고 통계적인 중복성을 제거할 필요가 있다. MPEG과 같은 동영상 부호화 표준에서는 DCT 변환과 움직임 보상을 통해 각각 공간적인 중복성과 시간적인 중복성을 제거한다. 움직임 보상은 현재 부호화되는 프레임과 참조 프레임 간의 움직임 예측을 통해 수행된다. 즉, 움직임 예측을 통해 얻어진 움직임 벡터는 참조 프레임으로부터 물체가 얼마만큼 움직였는지를 나타낸다. 그리고 통계적인 중복성은 엔트로피(entropy) 부호화에 의해 제거된다<sup>[4-6]</sup>. 즉, H.264/AVC의 P 프레임(predictive-coded frame) 내에서 움직임 보상에 의해 얻어진 움직임 벡터는 Exp-Golomb(Exponential Golomb) 부호화를 통해 통계적 중복성이 제거된다.

따라서 본 논문에서는 동영상 검색을 위한 특징을 추출하기 위해 Microsoft Visual C++로 제작한 H.264/AVC 스트림 파서(stream parser)를 사용하여 그림 1에 나타난 것과 같은 간단한 과정을 통해

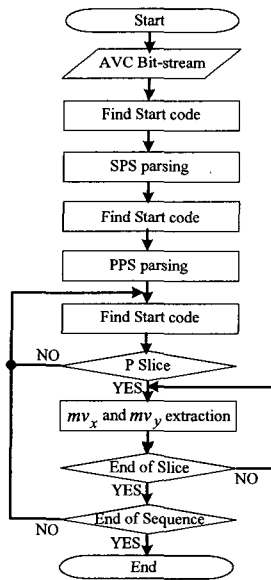


그림 1. 움직임 벡터 추출을 위한 블록도  
Fig. 1. Block diagram for motion vector extraction.

H.264/AVC로 부호화된 비트 스트림 내에서 움직임 벡터( $mv_x$  및  $mv_y$ )만을 추출하였다.

2. 동영상 검색을 위한 특징 추출

본 논문에서는 H.264/AVC 비트 스트림으로부터 추출한 움직임 벡터의 크기와 방향 성분을 그림 2와 같이 각각  $m$ 과  $\theta$ 로써 표현하였다.

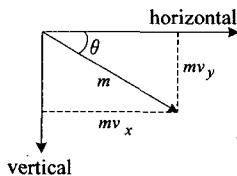


그림 2. 움직임 벡터의 크기와 방향  
Fig. 2. Magnitude and angle of a motion vector.

이 때,  $mv_x$ 와  $mv_y$ 를 각각 가로축과 세로축으로의 움직임 벡터 성분이라 하면 크기  $m$ 과 방향  $\theta$ 는  $mv_x$ 와  $mv_y$ 에 의해 수식 (1)과 같이 계산된다.

$$m = \sqrt{(mv_x^2 + mv_y^2)}$$

$$\theta = \arccos \frac{mv_x}{m} \tag{1}$$

가. 움직임 보상 블록에 따른 문제 해결

H.264/AVC는 움직임 보상의 정확도를 높이기 위해 다양한 블록 크기의 움직임 예측과 보상을 함에 따라 그림 3에서 보는 바와 같이 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 픽셀 블록 단위의 움직임 벡터를 갖게 된다.

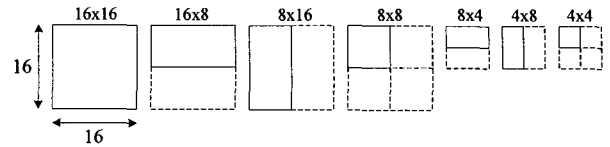


그림 3. H.264/AVC의 움직임 벡터의 단위 크기  
Fig. 3. Unit sizes of motion vector in H.264/AVC.

이러한 움직임 보상 특성으로 인해 같은 크기와 방향 성분을 가진 움직임 벡터라 하더라도 각각의 블록 크기에 따라 움직임 벡터가 나타내는 움직임 정도는 달리 표현된다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 구현의 용이성과 계산량 측면에서 가장 간단한 방법인, 다양한 크기의 움직임 벡터를 움직임 보상의 최소 단위인 4x4 픽셀 단위로 분할하고, 각 블록에 기존의 움직임 벡터와 동일한 값을 할당하는 방법을 사용하였다. 예를 들어, 움직임 벡터가 16x16 픽셀 블록일 경우 4x4 픽셀 블록 16개로 분할하고, 모든 4x4 픽셀 블록에 기존 16x16 픽셀 블록의 움직임 벡터 값을 할당한다. 또한, 8x8 픽셀 블록일 경우는 동일한 움직임 벡터 값을 갖는 4x4 픽셀 블록 4개로 분할한다. 실제 움직임 벡터의 정규화에 대한 구현은 매크로 블록(macro block) 단위로 움직임 벡터를 카운트(count)하는 과정에 포함되므로 블록 크기에 따라 기존의 움직임 벡터 값을 비트 시프트(bit shift) 연산하는 것만으로 처리될 수 있기에 계산량에 있어서의 이점이 있다.

나. 움직임 벡터의 양자화

앞서 가 절에서 언급한 바와 같이 프레임 내의 모든 움직임 벡터는 각각의 블록 크기에 따라 4x4 픽셀 블록으로 분할된다. 그리고 분할된 4x4 픽셀 블록의 움직임 벡터는 방향  $\theta$ 에 따라 수식 (2)와 같이 0°, 45°, 90°, 135°의 4 방향으로 양자화되어 해당 프레임에 대한 4 방향으로의 움직임 특성을 대표한다.

$$\hat{\theta} = \begin{cases} 0^\circ, & \text{if } 337.5^\circ < \theta \leq 22.5^\circ \\ 45^\circ, & \text{if } 22.5^\circ < \theta \leq 67.5^\circ \\ 90^\circ, & \text{if } 67.5^\circ < \theta \leq 112.5^\circ \\ 135^\circ, & \text{if } 112.5^\circ < \theta \leq 157.5^\circ \end{cases} \tag{2}$$

이 때,  $\hat{\theta}$ 는 0°, 45°, 90°, 135°의 양자화되는 방향을 나타내며, 그림 4는 4x4 픽셀 블록 단위로 분할된 각각의 움직임 벡터 방향 성분의 양자화에 대한 것이다.

그림 4에서  $f_0^i(n)$ ,  $f_{45}^i(n)$ ,  $f_{90}^i(n)$ ,  $f_{135}^i(n)$ 는 각각  $n$ 번째 프레임 내  $i$ 번째 4x4 픽셀 블록의 0°, 45°, 90°, 135°의 움직임 벡터 방향 성분의 양자화 값을 나타낸다.

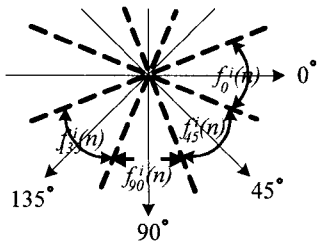


그림 4. 움직임 벡터 방향 성분의 양자화  
Fig. 4. Quantization of the angle of the motion vector.

135° 방향으로 양자화된 움직임 벡터의 크기를 나타낸다. 예를 들어, 움직임 벡터의 방향  $\theta$ 가 22.5°와 67.5° 사이에 있으면 움직임 벡터의 양의 크기를 갖는  $f_{45}^i(n)$ 로 양자화 되고, 200.5°와 245.5° 사이에 있으면 움직임 벡터의 음의 크기를 갖는  $f_{45}^i(n)$ 로 양자화된다.

다. 영역 기반의 특징 정보 추출

해당 프레임 내의 모든 4x4 픽셀 블록의 움직임 벡터는 각각의 움직임 벡터 방향 성분에 따라 4 방향으로 양자화된 후 동일한 방향 성분끼리 더해지고, 전체 4x4 픽셀 블록의 개수로 나누어져 해당 프레임의 4 방향으로의 움직임 정도를 표현하는 특징 정보로 사용된다. 그러나 이러한 프레임 기반의 움직임 정보는 한 프레임 내에서 빠르게 움직이는 물체나 복잡한 영상 내의 움직임을 대표하기는 힘들다. 따라서 이러한 문제를 해결하기 위해 본 논문에서는 그림 5와 같이 동영상 내의 모든 프레임에 대해 4개의 영역으로의 분할함으로써 영역 기반의 움직임 특징 정보를 추출하였다.

$R_0, R_1, R_2, R_3$ 를 각각 부 영상 0, 1, 2, 3 내 4x4 픽셀 블록의 집합이라 하고,  $F_{\theta}^i(k, n)$ 를  $k$ 번째 부 영상 내에서  $\hat{\theta}$  방향을 가지는 움직임 벡터의 평균 크기라고 하면  $F_{\theta}^i(k, n)$ 는 수식 (3)과 같이 계산된다.

$$F_{\theta}^i(k, n) = \frac{1}{N_{R_k}} \sum_{i \in R_k} f_{\theta}^i(n), \quad k = 0, 1, 2, 3 \quad (3)$$

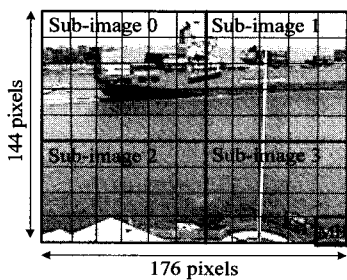


그림 5. 4개의 부 영상  
Fig. 5. Four sub-images.

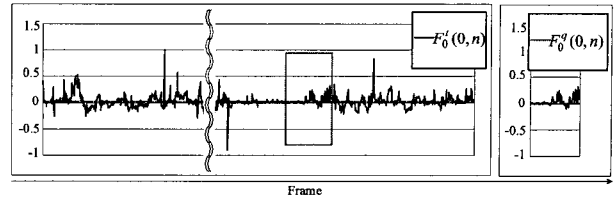


그림 6. (좌) 실험 동영상의  $F_0^t(0, n)$ , (우) 질의 동영상의  $F_0^q(0, n)$   
Fig. 6. (left)  $F_0^t(0, n)$  of test video, (right)  $F_0^q(0, n)$  of query video shot.

이 때,  $N_{R_k}$ 는  $k$ 번째 부 영상 내 전체 4x4 픽셀 블록의 개수를 나타낸다. 따라서 실험 동영상과 질의 동영상 내 두 동영상 샷 간의 유사도는 두 동영상 샷의 모든 프레임에 대해 계산되어 나열된  $F_{\theta}^t(k, n)$ 과  $F_{\theta}^q(k, n)$ 의 비교를 통해 측정된다. 그림 6은 두 동영상으로부터 계산된 0번째 부 영상 내 0° 방향의  $F_0^t(0, n)$ 와  $F_0^q(0, n)$ 의 예를 나타내고 있다. 이 때 위첨자 "t"와 "q"는 각각 실험 동영상과 질의 동영상을 의미한다.

움직임 보상 과정에서 생략되는 매크로 블록(skipped macro block)이나 인트라 매크로 블록이 발생하는 경우는 참조 프레임과 현재 프레임의 대응하는 블록 간에 움직임이 없거나 움직임 예측 범위를 벗어난 경우이다.<sup>[4]</sup> 따라서 이러한 경우는 이 장에서 설명한 동영상 검색을 위한 특징을 추출하는 과정에서 무시한다.

III. 유사도 측정

본 논문에서는 유사도 측정을 위해 실험 동영상과 질의 동영상 샷에서 추출한 특징들 간의 상호 상관계수<sup>[9]</sup>를 이용하였다. 그림 7에 나타난 바와 같이, 질의 동영상 샷은 한 프레임씩 우측으로 이동하며 실험 동영상의 모든 프레임에 대해 상호 상관계수를 계산하게 된다. 실험 동영상과 질의 동영상 샷 간의 유사도는 그림 8에

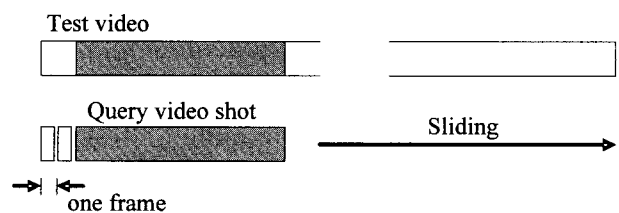


그림 7. 실험 동영상과의 상호 상관계수를 계산하기 위해 이동되는 질의 동영상 샷  
Fig. 7. Sliding query video shot for the computation of the cross-correlation between test video.

나타낸 바와 같이 각각의 부 영상에 해당하는 상호 상관계수에 의해 측정된다. 그림 8에서 질의 동영상 샷은 실험 동영상의  $l$ 번째 프레임으로부터 상호 상관계수를 측정한다고 가정하였고, 질의 동영상 샷의 전체 프레임 수는  $q_{\max}$ 이다. 따라서 해당하는 두 샷 간의 상호 상관계수의 계산을 위해 질의 동영상 샷 전체와  $l$ 번째 프레임부터  $(l+q_{\max})$ 번째 프레임까지의 실험 동영상이 사용된다. 그림 8에서  $C_{\hat{\theta}}^l(k)$ 은  $l$ 번째 프레임을 시작으로 하는 실험 동영상과 질의 동영상 샷의  $k$ 번째 부 영상 내  $\hat{\theta}$  방향을 가지는 특징들 간의 상호 상관계수를 나타낸다. 즉, 두 동영상 내 부 영상 샷 간의 상호 상관계수는  $C_{\hat{\theta}}^l(0)$ ,  $C_{\hat{\theta}}^l(1)$ ,  $C_{\hat{\theta}}^l(2)$ ,  $C_{\hat{\theta}}^l(3)$ 의 값을 가지고, 이 때  $C_{\hat{\theta}}^l(k)$ 는 수식 (4)와 같이 계산된다.

$$C_{\hat{\theta}}^l(k) = \frac{\sum_{j=1}^{q_{\max}} ((F_{\hat{\theta}}^l(k, l+j-1) - \overline{F_{\hat{\theta}}^l(k)}) (F_{\hat{\theta}}^q(k, j) - \overline{F_{\hat{\theta}}^q(k)}))}{\sqrt{\sum_{j=1}^{q_{\max}} (F_{\hat{\theta}}^l(k, l+j-1) - \overline{F_{\hat{\theta}}^l(k)})^2} \sqrt{\sum_{j=1}^{q_{\max}} (F_{\hat{\theta}}^q(k, j) - \overline{F_{\hat{\theta}}^q(k)})^2}} \quad (4)$$

수식 (4)에서  $\overline{F_{\hat{\theta}}^l(k)}$ 과  $\overline{F_{\hat{\theta}}^q(k)}$ 는

$$\overline{F_{\hat{\theta}}^l(k)} = \frac{1}{q_{\max}} \sum_{j=1}^{q_{\max}} F_{\hat{\theta}}^l(k, l+j-1), \quad (5)$$

$$\overline{F_{\hat{\theta}}^q(k)} = \frac{1}{q_{\max}} \sum_{j=1}^{q_{\max}} F_{\hat{\theta}}^q(k, j) \quad (6)$$

과 같이 계산된다. 수식 (4)에 의해 계산된 각각의 부 영상 별, 그리고 양자화된 방향  $\hat{\theta}$ 에 따른 상호 상관계수는 수식 (7)과 같이 모든 부 영상과 방향에 대해 평균을 나타내게 된다.

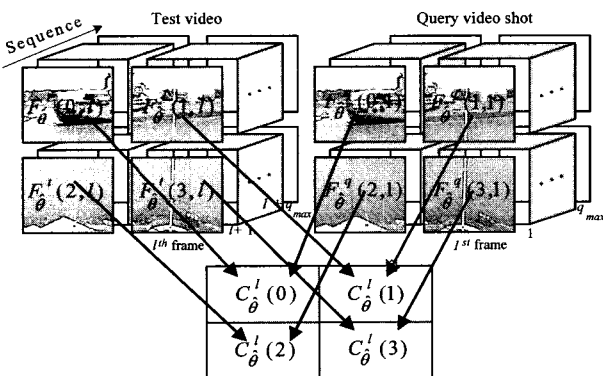


그림 8. 실험 동영상과 질의 동영상 간의 부 영상별 상호 상관계수

Fig. 8. Cross-correlations of each sub-image between test video and query video shot.

$$C_l = \frac{1}{16} \sum_{k=0}^3 (C_0^l(k) + C_{45}^l(k) + C_{90}^l(k) + C_{135}^l(k)) \quad (7)$$

여기서 16은 4 방향( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ )과 4개의 부 영상 (그림 8)의 모든 경우에 대한 평균 계산을 위한 값이다. 따라서 수식 (7)과 같이 계산되어진 상호 상관계수  $C_l$ 을 이용하여 다음 수식 (8)과 같이 실험 동영상 내에서 질의 동영상 샷과 가장 유사한 샷의 첫 번째 프레임을 찾게 된다. 이 때,  $n_{\max}$ 는 실험 동영상의 전체 프레임 수를 나타낸다.

$$l^* = \arg \max_{l=1, \dots, n_{\max}} C_l, \quad (8)$$

#### IV. 실험 및 결과

실험에 사용된 동영상은 MPEG 표준 동영상과 일반 동영상으로 구성되어 있다. 표 1과 2에 나타낸 바와 같이 실험에 사용된 동영상은 표준 동영상과 일반 동영상으로 구성하였고, 일반 동영상은 동영상의 내용에 따라 뉴스(news), 스포츠(sports), 레이스(racing), 비행신(scene)으로 구성되어 있다. 모든 동영상은 H.264/AVC로 부호화되었고, 부호화 파라미터(parameter)는 베이스라인 프로파일(baseline profile)<sup>[4]</sup>을 기반으로, 30fps (frame per second)의 프레임 율(frame-rate) 및 초당 3번의 인트라 프레임(즉, GOP=10)을 포함하도록 하였다. 또한, 실험 동영상과 질의 동영상 샷의 비트율(bit-rate)이 다른 경우에 대한 적용 여부를 판단하기 위해 질의 동영상은 일률적인 비트율인 300Kbps로, 실험 동영상은 50Kbps, 100Kbps, 200Kbps, 300Kbps, 400Kbps로 각각 부호화하여 실험하였다.

제안한 동영상 검색 방법의 성능은 ANMRR (Average Normalized Modified Retrieval Rank)<sup>[10]</sup>을

표 1. 실험에 사용된 동영상 (표준 동영상)  
Table 1. Test set (standard videos).

Sequence	Resolution	# of videos	# of frames
Test video	176x144	12	10558
Query video	176x144	12	800

표 2. 실험에 사용된 동영상 (일반 동영상)  
Table 2. Test set (commercial videos).

Sequence	Resolution	# of videos	# of frames
Test video	352x240	32	48161
Query video	352x240	8	1850

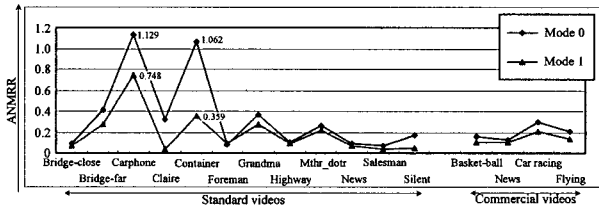


그림 9. Mode 0와 Mode 1의 검색 성능 비교  
Fig. 9. Performance comparison of Mode 0 and Mode 1.

표 3. 실험 동영상에 대한 검색 성능 평가  
Table 3. Performance evaluation with test sets.

Videos	ANMRR (Mode 0)	ANMRR (Mode 1)
Standard	0.348	0.193
Commercial	0.199	0.136

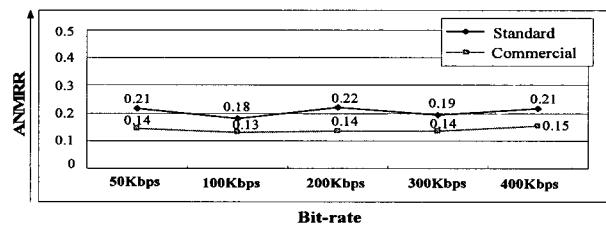


그림 10. 비트율의 변화에 따른 ANMRR 측정 결과  
Fig. 10. Variation of ANMRR with bit-rate increase.

통해 검증하였고, 그림 9와 표 3에 제안한 방법을 이용해 실험한 동영상 검색 결과를 나타내었다. 실험에 사용된 모든 동영상은 장면 전환이 없고, ANMRR 측정을 위해 각 동영상을 일률적인 단위로 8등분 하여 Ground-truth<sup>[10]</sup>로 설정하였다. 그림 9와 표 3에서 보는 바와 같이 영역 기반의 특징 정보를 이용한 방법(Mode 1)이 프레임 기반의 특징 정보를 이용한 방법(Mode 0)보다 향상된 성능을 보이는 것을 알 수 있다. 또한, 실험결과를 통해 본 논문에서 제안한 동영상 검색 방법을 이용하여 사용자의 요구, 즉, 본 실험에서의 질의 동영상과 유사한 움직임 특성을 나타내는 동영상을 효과적으로 검색할 수 있음을 확인하였다.

그림 10은 실험 동영상의 비트율을 변화시켜가며 실험한 결과를 보이고 있다. 그림 10에서 보는 바와 같이 본 논문에서 사용한 특징은 4영역으로 분할된 동영상 사이의 통계적인 움직임 변화량을 이용하였기에 실험 동영상과 질의 동영상 샷 사이의 비트율 차이로 인한 움직임 벡터의 변화는 검색 결과에 크게 영향을 주지 않음을 확인하였다.

그림 9의 Carphone.264와 Container.264 동영상의 경우 빠른 움직임을 가진 작은 물체에 의해 검색 성능이 다소 떨어지는 것을 확인할 수 있었다. 그림 11은 제안

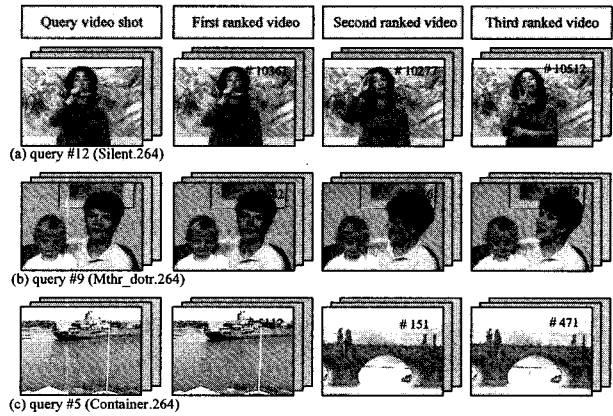


그림 11. 제안한 방법을 이용한 동영상 검색 결과  
Fig. 11. Video retrieval results with proposed method.

한 동영상 검색 방법을 이용한 검색 결과를 나타내고 있다. 각 영상은 질의 동영상과 실험을 통해 검색된 동영상의 첫 번째 프레임에 나타내고 있다. 그림 11의 (c)는 앞서 언급한 바와 같이 빠르게 변화하는 작은 물체를 가진 동영상의 검색 결과를 나타내었고, (a)와 (b)의 경우에 비해 검색 성능이 떨어지는 것을 확인할 수 있다.

## V. 결론

본 논문은 H.264/AVC 압축 영역에서의 새로운 동영상 검색 방법을 제안하였다. 제안한 방법은 H.264/AVC 비트 스트림으로부터 추출한 움직임 벡터만을 이용하였다. 그리고 실험 동영상과 질의 동영상 샷 간의 유사도를 측정하기 위해 부 영상 샷에서 추출한 특징들 간의 상호 상관계수를 이용하였다. 제안한 방법을 이용하여 H.264/AVC로 부호화된 동영상에 대해 실험한 결과 질의 동영상과 유사한 동영상을 검색할 수 있었고, 10558 프레임의 표준 동영상과 48161 프레임의 일반 동영상에 대해 0.2 이하의 ANMRR 결과를 나타내었다.

## 참고 문헌

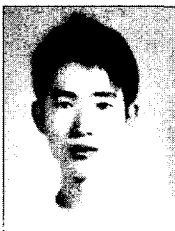
- [1] W.E. Farag and H.A. Wahab, "A human-based technique for measuring video data similarity," in Proc. of IEEE on International Symposium on Computers and Communication, Vol. 2, pp. 768-774, 2003.
- [2] Y. T. Kim and T.S. Chua, "Retrieval of news video using video sequence matching," in Proc. of IEEE Conf. on Multimedia Modeling, pp. 68-75, 2005.
- [3] J.F. Chen, H.Y. Mark Liao and C.W. Lin, "Fast

- video retrieval via the statistics of motion," in Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing, Vol. 2, pp. 437-440, 2005.
- [4] E. Ardizzone, M. La Cascia, A. Avanzato and A. Bruna, "Video indexing using MPEG motion compensation vectors," in IEEE international Conference on Multimedia Computing and Systems, Vol. 2, pp. 725-729, 1999.
- [5] I.E.G. Richardson, *H.264 and MPEG-4 video Compression*, John Wiley & Sons, 2003.
- [6] T. Wiegand and G. Sullivan, "Draft ITU-T recommendation and final draft international standard of joint video specification," Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, 2003.
- [7] A. Tamhankar and K.R. Rao, "An overview of H.264/MPEG-4 Part 10," 4<sup>th</sup> EURASIP Conf. on Video/Image Processing and Multimedia Communications, 2003.
- [8] S.-S. Cheung and A. Zakhor, "Efficient video similarity measurement with video signature," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, pp. 59-74, 2003.
- [9] K. Konstantinos and T. Sergios, *Pattern recognition*, Academic press, 2003.
- [10] B.S. Manjunath, A. Philippe and S. Thomas, *Introduction to MPEG-7 multimedia content interface*, John Wiley & Sons, 2002.

---

 저 자 소 개
 

---



변 주 완(정회원)  
 2005년 동국대학교 전자공학과  
 학사 졸업.  
 2005년~현재 동국대학교  
 전자공학과 석사과정  
 <주관심분야 : 영상압축, 영상검  
 색>



김 성 민(정회원)  
 2003년 동국대학교  
 전자공학과 학사졸업.  
 2005년 동국대학교  
 전자공학과 석사졸업  
 2005년~동국대학교  
 전자공학과 박사 과정.  
 <주관심분야 : Watermark, Retrieval, MPEG>



원 치 선(정회원)-교신저자  
 1982년 고려대학교 전자공학과  
 학사 졸업.  
 1986년 Univ. of Massachusetts  
 Amherst (공학석사)  
 1990년 Univ. of Massachusetts  
 Amherst (공학박사)  
 1989년~1992년 금성사 중앙연구소 선임연구원  
 1992년~현재 동국대학교 전자공학과 교수  
 <주관심분야 : 영상분할, 영상검색, 워터마킹>