

논문 2004-41SP-5-2

압축 영역에서의 양방향 예측 구조를 이용한 움직임 흐름 분석

(Motion Flow Analysis using Bi-directional Prediction-Independent Framework in MPEG Compressed Domain)

김 낙 우*, 김 태 용**, 최 종 수**

(Kim Nac-Woo, Kim Tae-Yong, and Choi Jong-Soo)

요 약

비디오 시퀀스는 일반적으로 다양한 움직임을 가지는 객체들로 구성되어 있기 때문에, 움직임 특징은 비디오 검색 등에서 매우 중요한 역할을 한다. 본 논문에서는 MPEG 압축 영상에서의 움직임 벡터를 비디오 영상의 움직임 표현 서술자로 활용하는 새로운 방법이 제안된다. 즉, 압축 영상에서의 다양한 움직임 벡터를 프레임이나 매크로블록 예측 구조에 관계없이 단일 움직임 방향만을 갖도록 하여, 이것을 해당 영상의 서술자로 활용한다. 이를 위하여, 본 논문에서는 양방향 예측 구조를 이용한 벡터 재해석 기법을 제안한다. 보통, 압축 영역에서의 각 프레임 움직임 해석 시, 움직임 벡터가 없는 I 프레임과 그 외 프레임들의 직접 비교는 불가능하지만, 제안 기법은 I, B, P 프레임 등의 모든 프레임에서 동등하게 벡터 해석을 할 수 있게 한다. 제안된 알고리즘은 압축 영상의 전체 복원과정 없이 매크로 블록 영역 상에서 처리함으로써 시간 손실을 줄이고 있으며, 실험 결과는 제안된 방법의 높은 성능을 잘 나타내어 주고 있다.

Abstract

Because video sequence consists of dynamic objects in nature, the object motion in video is an effective feature in describing the contents of video sequence and motion feature plays an important role in video retrieval. In this paper, we propose a method that converts motion vectors (MVs) to a uniform set on MPEG coded domain, independent of the frame type and the direction of prediction, and utilizes these normalized MVs (N-MVs) as motion descriptor to understand video contents. We describe a frame-type independent representation of the various types of frames presented in an MPEG video in which all frames can be considered equivalently, without full-decoding. In the experiments, we show that the proposed method is better than the conventional one in terms of performance.

Keywords: Motion analysis, Normalized motion vector, Bi-directional prediction-independent framework

I. 서 론

최근 멀티미디어 정보의 폭발적 증가와 맞물려 비디오 영상물의 통신, 저장, 처리 기술 또한 급격하게 발전하고 있다. 이에 따라 사용자에게 보다 효과적으로 멀

티미디어 자료 검색 서비스를 제공하기 위하여 비디오 데이터베이스 관리 시스템(Video Database Management System)이나 내용 기반 비디오 브라우저 등이 폭 넓게 연구되고 있으며 정지 영상의 검색을 지나 동 영상 검색에 이르기 까지 그 연구 분야가 점차로 확대되고 있다^{[1][2]}.

거대한 정보의 양과 이의 통제를 위한 관리 시스템의 필요성은 비디오 정보의 특징 표현을 요구하고 있으며, 비디오에서의 움직임 특징은 비디오 인덱싱이나 검색을 위하여 가장 중요한 요소로 고려된다. 이것이 오늘날 움직임 기반의 비디오 분석 기법이 활발히 연구되고 있는 이유이며, 이러한 움직임 기반의 비디오 분석의 주 쟁점은 움직임 계(界)로부터 간명하고 효율적인 움직임

* 학생회원, ** 정회원, 중앙대학교 첨단영상대학원 영상공학과
(Dept. of Image Engineering, Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University)
※ 본 연구는 과기부의 국가지정연구실(2000-N-NL-01-C-285)과 교육부의 두뇌한국 21 사업(BK21)의 연구지원으로 수행되었습니다.
접수일자: 2003년8월14일, 수정완료일: 2004년6월14일

서술자를 취득하는 것이다. 그러나, MPEG-1/2영상에서의 움직임 특징 추출은 압축 영역에서의 움직임 벡터를 이용하기 때문에 편리한 점은 있으나, 성긴 움직임 정보만을 추출한다는 점과 벡터 정보가 정규화 되어있지 않다는 점에서 문제가 발생한다.

본 연구에서는, MPEG 압축 영역에서의 움직임 벡터 정보를 양방향 예측 독립 구조(Bi-directional Prediction-Independent Framework : BPIF)를 이용한 움직임 분석 기법에 의해 정규화시킴으로써 효과적인 움직임 서술자의 활용 가능성에 대해 논하고, 이를 이용한 각 프레임에서의 정확한 움직임 해석을 제안하고 있다. 본 논문에서의 제안 방법은 성긴 움직임 벡터계를 보완하여 조밀하고 포괄적인 움직임 벡터계를 만드는 데에 그 목적이 있다. 이를 통해 정규화된 움직임 벡터 정보는 움직임 서술자로서의 높은 활용 가능성을 가지게 된다. 양방향 예측 구조를 통해 정규화된 움직임 벡터는 특히 비디오 시퀀스에서의 전역적인 카메라 움직임 검출과 비디오 인덱싱, 동영상 검색을 위한 특징 정보로서도 그 활용이 가능하다.

본 논문의 구성은 다음과 같다. II장에서는 움직임 해석에 대한 관련 연구를 다루고, III장에서 압축 영역에서의 움직임 해석을 위한 알고리즘을 제안한다. 그리고, IV, V장과 VI장에서 각각 알고리즘에 대한 검증, 응용 결과 및 그에 대한 결론을 도출한다. 그림 1은 본 연구의 대략적인 블록도를 나타낸 것이다.

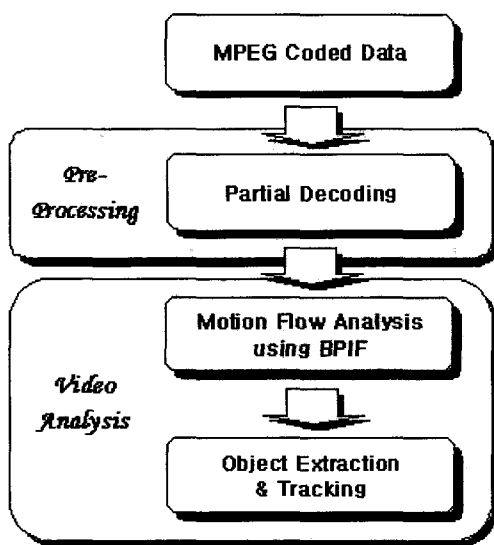


그림 1. 제안된 알고리즘의 블록도
Fig. 1. Block diagram of the proposed algorithm.

II. 관련 연구

움직임은 비디오의 내용을 이해하기 위한 중요 요소로서 컴퓨터 비전과 영상 처리 연구가 시작된 이래 움직임에 대한 예측 문제는 주요 주제로서 폭넓게 연구되고 있다. 움직임 예측이란 연속적인 두 개의 프레임으로부터 움직임 정보를 추출하는 일반적인 방법을 말하는데, 크게 파라미터 모델과 비-파라미터 모델(non-parameter)로 구분된다^[3]. 전역적인 파라미터 움직임 예측은 어파인(affine) 모델처럼 카메라 모션이나 주요 움직임의 파라미터 모델을 생성하는 것이고, 비-파라미터 모델에서의 움직임 예측은 광류(optical flow)와 같이 변위 쌍을 찾는 것이다. 이로부터 우리는 해당 프레임의 움직임 표현을 얻는다. 최근, 이러한 움직임 분석 모델을 적용한 내용 기반 비디오 검색 시스템이 활발히 연구되고 있다. 그러나, 전역적인 파라미터 모델을 이용한 움직임 서술자는 너무 성기고, 비-파라미터에서의 움직임 서술자는 지나치게 세밀하기 때문에, 각각 움직임 표현에서의 움직임 특징이 불충분해지거나 많은 시간 소비로 인하여 비효율적이 되기 쉽다는 단점이 있다. 이러한 이유로 인하여 최근 압축 영역에서의 움직임 벡터를 기반으로 한 비-파라미터 움직임 예측 기법이 활발히 연구되고 있다.

비-파라미터 움직임 예측 기법을 살펴보면, 움직임 기반의 비디오 인덱싱을 위하여 일반적으로 컴퓨터 비전 분야에서는 광류(optical flow)를 이용하고 영상 부호화 관련 분야에서는 블록 정합 기법 (Block Matching Algorithm) 등과 같은 움직임 분석 기법을 널리 사용하고 있다. Cherfaoui와 Bertin은 광류를 이용하여 카메라의 전역적인 움직임 매개 변수를 계산한 후 이를 이용하여 샷 내의 움직임을 분류하는 기법을 제안하였다^[4]. 또, Zhang은 [5]에서 블록 정합 기법을 기반으로 하여 움직임 벡터의 방향과 수렴 및 발산 정도를 계산함으로써 움직임 분류를 수행하기도 하였다. 그러나, 이와 같은 방법은 정합점을 계산하는 데에 있어 많은 계산량과 시간 소요가 필요하다는 단점이 있기 때문에, MPEG 비디오를 이용한 샷기반의 움직임 분류 기법은 대부분 P나 B프레임에서의 움직임 벡터를 이용한다^{[6][7]}. Kobla는 MPEG 비디오 상의 움직임 벡터를 프레임 종류에 상관없이 독립적인 형태로 표현하여 이를 이용하는 방법을 제안하였다^[8]. 그러나, 이 방법은 참조 프레임 등에서의 벡터 표현에 있어 단방향성만을 고려하는 단점을 나타내었다.

움직임 정보는 칼라나 모양, 질감 정보 같은 시각 특성의 시간적 변화 뒤에 항상 숨겨져 있기 때문에 비디오 검색에 있어 쉽게 이러한 움직임 정보를 사용하기는 어렵다. 따라서, 원 영상으로부터 움직임 정보를 추출하고, 그것을 움직임 표현의 정교한 형태로 바꾸어주는 것이 비디오 검색에 있어 필수적이다. 오늘날, [9,10,11]에서 제안된 움직임 표현들이나 MPEG 영역에서의 움직임 벡터계 등이 빠른 비디오 인덱싱이나 검색을 위해 사용되고 있다^{[12][13]}. 그러나, 이러한 많은 노력에도 불구하고, 움직임 분석이 내용 기반 비디오 분석이나 검색을 위한 효과적인 방법이 되기 위해서는 여전히 효과적인 벡터 표현을 위한 분석이 먼저 필요하다.

III. 움직임 벡터의 재추정을 이용한 움직임 해석연구

1. MPEG 시퀀스의 기본 구조

MPEG-1 스트림은 일반적으로 I 프레임, B 프레임, P 프레임 등의 세 가지의 유형을 갖는 GOP(group of pictures)의 시퀀스로 구성되어 있다. 이 중 I 프레임은 인트라(intra) 부호화 영상으로서 프레임간 예측을 쓰지 않고 생성되므로, 움직임 벡터가 존재하지 않는다. P 프레임은 하나의 레퍼런스 영상으로부터 예측 부호화되고, 매크로 블록이 가질 수 있는 최대의 움직임 벡터의 수는 하나이다. 또한, B 프레임은 양방향 영상으로부터 예측 부호화되기 때문에 하나의 매크로 블록이 두 개의 움직임 벡터를 가질 수도 있다^[14]. 압축 스트림에서의 매크로 블록은 같은 프레임 내에서조차 그 프레임의 유형에 따라 다른 수의 움직임 벡터를 가지게 되고 이러한 움직임 벡터는 그 예측 방향 또한 일률적이지 않다. 즉, 매크로 블록에서의 움직임 벡터를 이용하여 프레임에서의 움직임을 예측하고자 할 때에는 좀 더 효율적인 움직임 해석을 위하여 일반적인 매크로블록에서의 벡터 정보를 프레임의 유형과 예측 방향 등에 무관한 형태의 정규화된 움직임 벡터로 전환하는 작업이 먼저 선행되어야 한다. 이 논문에서는, 현재 프레임에서의 각각의 움직임 벡터를 다음 프레임 대응 블록에서의 역방향 벡터로 일괄적으로 표현함으로써 전체의 프레임 시퀀스를 역방향 벡터만을 가지는 프레임들로 재구성하였다.

2. 움직임 벡터 재추정 방법

MPEG 시퀀스의 GOP구조상에서 두 개의 연속적인 참조 프레임을 R_i 와 R_j 라 하고, 이 참조 프레임들 사이의 B 프레임들을 B_1, \dots, B_n (n 은 두 참조 프레임 사이의 B

프레임의 수)라고 한다면, 이러한 프레임들 간의 유기적 관계를 통해 전체 시퀀스의 움직임을 역방향 움직임 벡터만을 갖는 프레임 시퀀스로 재 유추할 수 있다^{[8][15]}. 이 방법은 크게 R_i, B_n 프레임에서의 움직임 재추정 방법과 $B_1 \sim B_{n-1}$ 프레임에서의 움직임 재추정 방법으로 구분된다.

가. Motion flow for R_i and B_n frame

일반적으로 R_i 프레임에서의 움직임 벡터 추정에는 참조 프레임 R_i 와 B_1 프레임 간의 순방향 예측된 움직임 벡터를 이용한다. B_n 프레임에서의 벡터 추정은 참조 프레임 R_j 와 B_n 프레임 간의 역방향 예측된 움직임 벡터를 이용한다. B_1 프레임에서의 매크로 블록 $B_{1(u,v)}$ (u, v 는 해당 프레임의 매크로 블록들 중에서 현재 매크로 블록의 위치를 표시:그림 2)가 움직임 벡터 (x, y) 를 갖는다는 의미는, R_i 프레임에서의 매크로 블록 $R_{i(u,v)}$ 가 B_1 프레임으로 움직임 벡터 (x, y) 만큼 이동했다는 것으로 표현될 수 있고, 이로부터 참조 프레임 R_i 에서의 움직임 추정 벡터를 얻는다. B_n 프레임에서의 벡터 추정 역시 이와 같은 방식을 이용하여 구할 수 있다^[8].

그러나, GOP 구조에서의 이러한 벡터 추정 방식은 각각의 프레임을 나타내기에 충분할 만큼의 유효 움직임 벡터를 추출하지 못하다는 문제점을 가지고 있다. 이러한 방식은 B_n 프레임이 비단 R_i 프레임에서의 역방향 움직임 벡터뿐만 아니라 R_i 프레임으로부터 순방향 움직임 벡터 또한 가질 수 있다는 점, 그리고 R_i 프레임에서의 벡터 추정 시 또한 위와 마찬가지로 B_1 프레임이 순방향 움직임 벡터만이 아닌 R_j 프레임으로부터의 역방향 움직임 벡터를 가질 수도 있다는 점 등이 이전 방법에서는 고려되지 않고 있다.

제안된 방법에서는 그림 3와 같이 R_i 프레임으로부터 R_j 프레임으로의 순방향 벡터로부터 R_i 프레임과 B_n 프레임에서의 벡터 예측을 보조함으로써 어느 한 방향에서의 벡터 예측이 아닌 양방향 예측을 가능하게 한다.

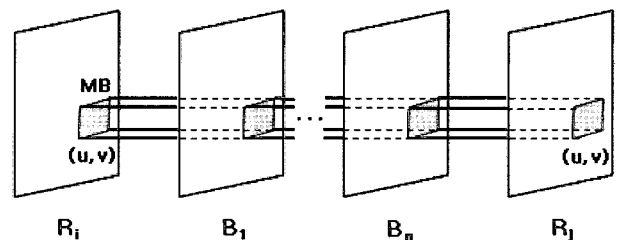
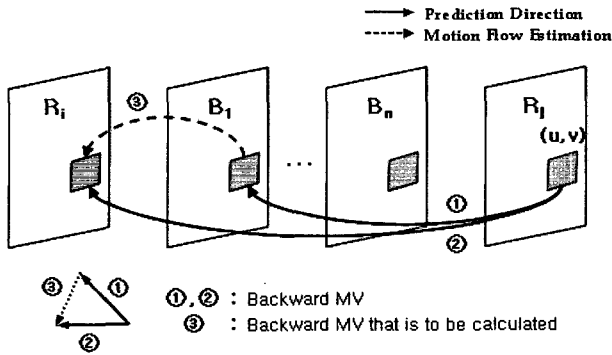
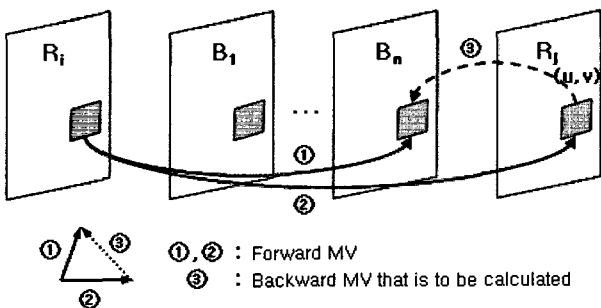


그림 2. u, v 좌표에서의 대응 매크로 블록 표현
Fig. 2. Illustration of MB on u, v coordinates.



(a) R_i 프레임에서의 움직임 재추정 기법
(a) Flow estimation of MVs in R_i frames.



(b) B_n 프레임에서의 움직임 재추정 기법
(b) Flow estimation of MVs in B_n frames.

그림 3. R_i 와 B_n 프레임에서의 움직임 재추정 기법
Fig. 3. Flow estimation of MVs in R_i and B_n frames.

즉, R_i 프레임 상의 벡터 추정을 위해 B_1 프레임에서의 순방향 벡터만을 사용한 기존의 방법과 달리 R_i 프레임에서 B_1 프레임으로의 역방향 벡터와 R_i 프레임에서 R_i 프레임으로의 순방향 벡터를 이용한 벡터 재해석 방법을 이용함으로써 R_i 프레임에서의 예측 벡터의 수를 크게 늘릴 수 있다. B_n 프레임에서의 벡터 추정 또한 이와 같은 방법을 이용하여 구할 수 있다^[16]. 그림 3-(a)는 R_i 프레임에서의 움직임 재추정을 위한 제안 방법을 보여주고 있으며 수식 (1)로 표현된다.

$$\overline{R_i R_j} = \overline{R_i B_1} + \overline{B_1 R_j} \quad (1)$$

B_n 프레임에서의 움직임 재추정 기법은 그림 3-(b)와 수식 (2)로서 표현된다.

$$\overline{R_i R_j} = \overline{R_i B_n} - \overline{B_n R_j} \quad (2)$$

나. Motion flow for $B_1 \sim B_{n-1}$ frame

연속적인 B 프레임에서의 벡터 추정은 해당 프레임에서의 레퍼런스 프레임에 대한 움직임 벡터와 그에 대응하는 매크로블록을 분석함으로써 이루어진다. 각 B

프레임에서의 MB는 순방향 예측 타입(F), 역방향 예측 타입(B), 혹은 양방향 예측 타입(D) 등의 세가지 예측 타입을 가질 수 있기 때문에, 아홉 개의 예측 조합이 가능하다. 이를 우리는 FF, FB, FD, BF, BB, BD, DF, DB, and DD로 표현한다. 이들 아홉 개의 조합은 레퍼런스 프레임에 대한 움직임 벡터를 분석함으로써 각각 개별적으로 고려되어야 한다. 알고리즘은 연속된 프레임에서의 대응 매크로 블록 조합에 따라 다음의 네 경우로 나뉜다.

- 경우 1) 대응 매크로 블록이 모두 순방향 벡터를 가지는 경우: FF, FD, DF, DD ;
- 경우 2) 대응 매크로 블록이 모두 역방향 벡터를 가지는 경우 : BB, BD, DB ;
- 경우 3) 대응 매크로 블록이 순방향 + 역방향 벡터를 가지는 경우 : FB ;
- 경우 4) 대응 매크로 블록이 역방향 + 순방향 벡터를 가지는 경우 : BF

이 중 경우 1과 2에서의 벡터 추정은 R_i 나 B_n 프레임에서의 벡터 추정 방식과 유사한 방법으로 쉽게 얻을 수 있다. 경우 1에 대한 그림 설명은 그림 4-(a)에 나타나고 있으며, 수식 표현은 (3)과 같다. 그림 4-(b)와 수식 (4)는 경우 2에 해당하는 그림과 수식을 보여주고 있다.

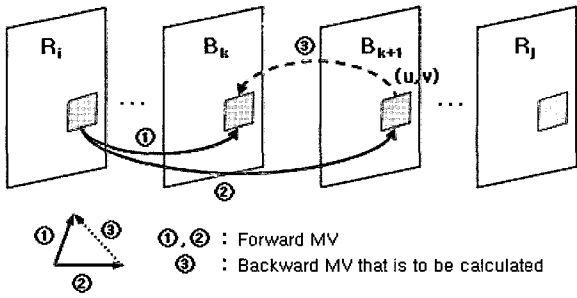
$$-\overline{B_{k+1} R_i} \equiv \overline{R_i B_k} + \overline{B_k B_{k+1}} \quad (3)$$

$$\overline{B_k R_j} \equiv \overline{B_k B_{k+1}} + \overline{B_{k+1} R_j} \quad (4)$$

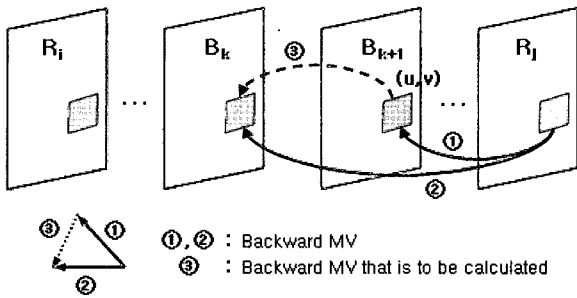
경우 3과 4에서의 움직임 추정을 위해서는 R_i 프레임으로부터 R_i 프레임으로의 순방향 벡터 $\overline{R_i R_j}$ 를 이용한다. FB 프레임에서의 추정 벡터 $\overline{B_k B_{k+1}}$ 은 기존의 대응 매크로 블록에서의 움직임 벡터 $\overline{B_{k+1} R_j}$, $\overline{R_i B_k}$ 와 순방향 움직임 벡터 $\overline{R_i R_j}$ 를 이용하여 얻는다. 그림 4-(c)는 이러한 알고리즘을 보여주고 있으며, 수식 (5)로 표현된다.

$$\overline{R_i R_j} - \overline{R_i B_k} = \overline{B_k B_{k+1}} + \overline{B_{k+1} R_j} \quad (5)$$

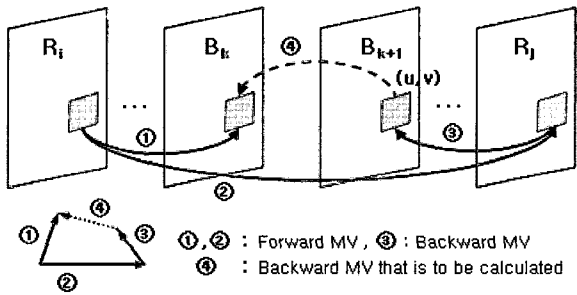
또한, 경우 4과 같은 BF 조합 프레임에서의 움직임 재추정은 $\overline{B_n B_n}$ 와 $\overline{B_n B_n}$ 를 이용하여 그림 4-(d)와 같이 추정 벡터 $\overline{B_n B_n}$ 을 얻는다. 그림으로부터 수식 (6), (7)



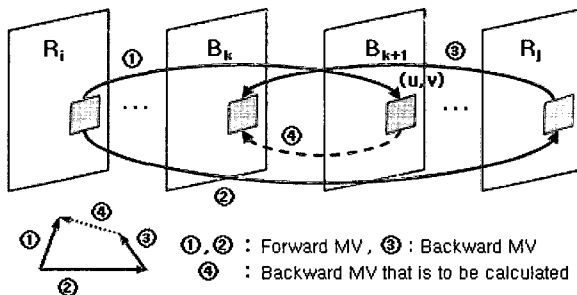
(a) 경우 1에서의 움직임 재추정 기법
(a) Flow estimation of MVs in case 1.



(b) 경우 2에서의 움직임 재추정 기법
(b) Flow estimation of MVs in case 2.



(c) 경우 3에서의 움직임 재추정 기법
(c) Flow estimation of MVs in case 3.



(d) 경우 4에서의 움직임 재추정 기법
(d) Flow estimation of MVs in case 4.

그림 4. B₁ ~ B_{n-1} 프레임에서의 움직임 재추정 기법
Fig. 4. Flow estimation of MVs in B₁ ~ B_{n-1} frames.

을 얻을 수 있으며 이로부터 수식 (8)의 유추가 가능하다.

$$-R_j R_i = R_j B_k + B_k R_j \quad (6)$$

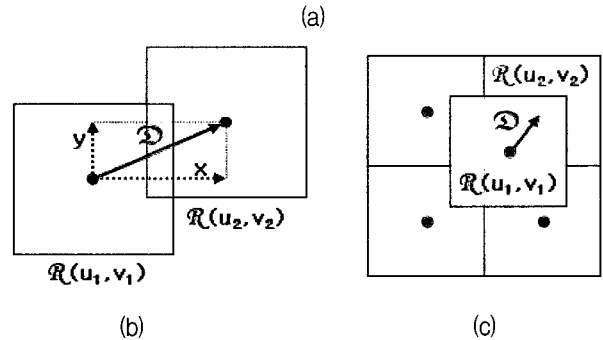
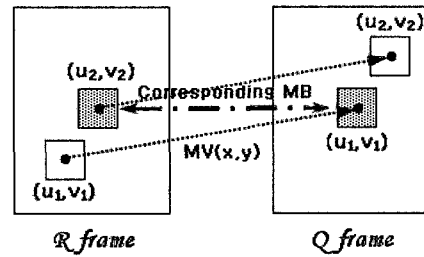


그림 5. 오차 추정 (a) 대응 매크로블록의 오류 (b) 오차 거리 (c) 매크로블록 매칭
Fig. 5. Error estimation. (a) MB mismatching. (b) Error distance. (c) MB matching.

$$\overrightarrow{R_i B_{k+1}} = \overrightarrow{B_k B_{k+1}} + \overrightarrow{R_i B_k} \quad (7)$$

$$\overrightarrow{R_i R_j} - \overrightarrow{R_i B_{k+1}} = \overrightarrow{B_k R_j} - \overrightarrow{B_k B_{k+1}} \quad (8)$$

그림에서 실선은 실제적인 매크로 블록의 움직임 벡터 방향을 표시한 것이고, 점선은 이를 통해 재추정된 움직임 벡터의 방향을 표시하고 있다.

3. 제안하는 움직임 재추정 방법의 적용 범위

본 논문의 움직임 추정 방법은 '피 참조프레임 Q(B or P 프레임)에서의 매크로블록 Q(u₁,v₁) 이 움직임 벡터 (x,y)를 가졌다는 것은 참조프레임 R(I or P프레임)에서의 매크로 블록 R(u₂,v₂) 가 Q프레임으로 움직임 벡터 (x,y)만큼 이동한 것과 동일하다'는 가정에서 출발한다. 이는 그림 5-(a)에서와 같이 Q(u₁,v₁) 에서의 움직임 벡터(x,y)는 R(u₂,v₂) 가 아닌 R(u₁,v₁)으로부터의 움직임 벡터라는 점에서 문제가 된다. R(u₂,v₂) 가 Q프레임으로 움직임 벡터 (x,y)만큼 이동한 것은 실제로 Q(u₂,v₂)이어야 한다. 따라서, 제안한 가정을 성립시키기 위해서는 그림 5-(b)에 도시하고 있는 오차거리 D가 최소화되어야 한다. 즉, 압축 영상에서의 움직임 벡터 대부분이 오차거리 이내에 존재해야 한다.

가정의 검증을 위해 우리는 다양한 영상으로부터 움직임 벡터 히스토그램을 얻고 이를 정규 분포로 나타내

보았다. 정규분포는 평균(μ)와 표준 편차(σ)에 의해 결정되는 분포로서 평균을 중심으로 종 모양의 확률밀도 함수($p(\mathcal{D})$)를 갖는다.

$$f(\mathcal{D}) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left[-(\mathcal{D} - M)^2 / 2\sigma^2\right]$$

for $-\infty < \mathcal{D} < \infty$

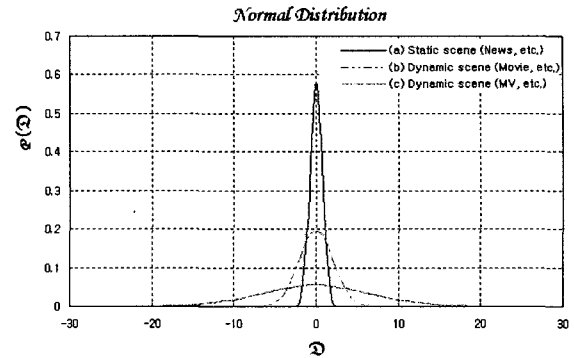
그림 6은 뉴스, 영화, 뮤직 비디오 등의 다양한 영상에서 얻은 정규분포를 보여주고 있으며, 표 1은 이의 수치값을 나타내고 있다. 우리는 최대오차거리 \mathcal{D} 를 블록 크기($-8 \leq \mathcal{D} \leq 8$)이하일 때와 매크로블록 크기($-16 \leq \mathcal{D} \leq 16$)이하일 때로 나누어 계산하였다. 최대오차거리 \mathcal{D} 가 블록 크기 이하일 경우 $Q_{(u1,v1)}$ 의 대응 매크로 블록 $R_{(u2,v2)}$ 는 $R_{(u1,v1)}$ 와 블록 크기 이하의 오차거리 내에 존재하게 되므로, $R_{(u2,v2)}$ 에 이웃한 어느 매크로 블록도 $R_{(u2,v2)}$ 만큼 $R_{(u1,v1)}$ 의 영역을 포괄하지 않는다는 것을 알 수 있다(그림 5-(c)). 이 경우 대응 매크로블록에 대한 우리의 초기가정은 매우 타당하다 할 수 있다.

표 1에서와 같이 대부분의 압축 영상은 블록 크기 이하의 움직임을 갖는다. 뮤직 비디오와 같이 빠른 움직임 영상이 아니라면 영상의 움직임 벡터가 블록 크기 이하일 확률은 99% 이상이다. 뮤직 비디오 영상의 경우, 중간 정도 빠르기의 움직임을 갖는 영상이라 할지라도 약 80%의 확률로서 움직임 벡터는 블록 이하의 움직임을 갖는다. 주목할 만한 것은, 실험 영상이 IBBPBBP구조의 시퀀스를 이용한 것이므로, 움직임 벡터의 크기를 프레임 단위로 환산할 경우, 최대 1/3까지 좀더 작아질 확률도 남아있다는 것이다. 이는 표준편차를 크게 감소시켜, 평균값 주위의 확률밀도 치를 크게 증가시킬 것이다.

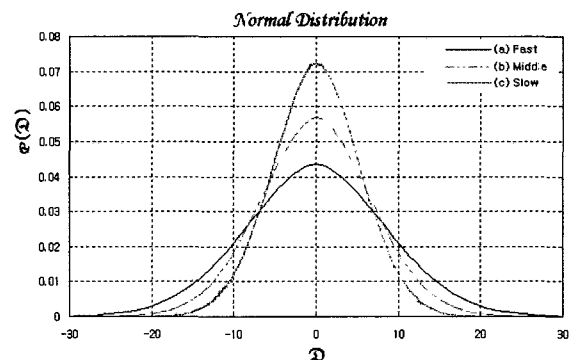
그러나, 빠른 움직임을 갖는 압축 영상의 경우 움직임 크기가 블록 이하일 확률이 70% 아래로 떨어지는 것을 표 1에서 확인할 수 있다. 즉, 움직임 벡터의 표준편차가 7이상을 갖는 빠른 움직임 영상에서는 제안한 알고리즘의 적용에 한계가 발생한다. 다만, 이 경우에 있어서도 그 움직임 크기가 매크로 블록 단위 이상을 벗어나지 않으므로, 영상의 국부 움직임 특성이 아닌 전역적 움직임 특성을 이해하는 데에 있어 제안하는 알고리즘의 이용은 충분히 가능하다.

IV. 결 증

이 장에서는 3장에서 제안된 움직임 벡터 추정 기법



(a)



(b)

그림 6. 정규 분포 (a) 다양한 영상 (b) 뮤직비디오
Fig. 6. Normal Distribution. (a) Various scene. (b) MV scene.

표 1. 정규 분포
Table 1. Normal Distribution.

Contents	Parameter	M	σ	$P(-8 \leq \mathcal{D} \leq 8)$	$P(-16 \leq \mathcal{D} \leq 16)$
News		0.03	0.69	0.99	0.99
Movies		0.05	2.04	0.99	0.99
Music Video	Fast	-0.98	9.05	0.68	0.94
	Middle	-0.53	6.99	0.79	0.98
	Slow	-0.36	5.48	0.85	0.99

에 대한 유효성 및 그 성능을 제시한다. 본 논문에서는 다양한 카메라의 움직임과 객체의 움직임을 가지고 있는 MPEG-1 스트림으로부터 제안된 알고리즘의 성능을 평가하였다.

1. 정규화된 움직임 벡터 수의 비교

우리는 각 프레임에서의 정규화된 움직임 벡터 수의 비교를 위해 IPP 부호화 프레임에서의 움직임 벡터를 기준 벡터계로 이용하였다. 실험은 동일한 시퀀스 영상에 대해 IPP 부호화와 IPB 부호화를 수행한 후, IPP 부호화 프레임에서의 움직임 벡터계를 이에 대응하는 IPB 프레임에서의 움직임 재추정 벡터계와 비교함으로써 제안된 알고리즘의 유효성을 검증하는 방식으로 이

루어졌다.

표 2에서 보듯 제안된 기법에서의 정규화된 움직임 벡터 수는 [8]에 비해 평균 18% 증가하는 것을 알 수 있다. 세분하여 볼 때, R_i 프레임에서의 벡터 수 증가는 8%, B_1 프레임에서의 벡터 수 증가는 7%, B_n 프레임에서의 벡터 수 증가는 44%로서 B_n 프레임에서의 벡터 수 증가가 매우 두드러지게 나타난다. 이러한 경향은 실험 영상의 B_n 프레임이 역방향의 벡터 성분을 많이 가지지 않을수록 더욱 심해지며, 이는 MPEG 시퀀스에서 대부분의 B_n 프레임은 R_i 프레임으로부터의 역방향 예측 벡터보다 오히려 R_i 프레임에서의 순방향 예측 벡터를 더 많이 가지고 있다는 것을 보여주는 예이다. 움직임 벡터 추출 비율은 IPP 프레임 시퀀스에서의 움직임 벡터 수와 IPB 프레임 시퀀스에서의 움직임 벡터 수 비교를 통하여 나타내었으며, 이러한 결과는 추출된 움직임 벡터가 IPP 프레임과 IPB 프레임의 해당 매크로 블록에서 서로 일치하는지의 여부가 고려되지 않은 것이다. 아래의 수식은 움직임 벡터 추출 비율을 계산하기 위한 식이다.

$$MV\ Detected\ Ratio = \frac{Number\ of\ MV\ in\ IPB\ frame}{Number\ of\ MV\ in\ IPP\ frame}$$

2. 정규화된 움직임 벡터의 유효성 검증

정규화된 움직임 추정 벡터 수의 증가가 바로 해당 추정 벡터의 유효성을 입증하는 것은 아니다. 우리는 각 매크로블록에서의 정규화된 추정 벡터가 적절한지의 여부를 검증하기 위하여 IPP 프레임과 IPB 프레임으로부터 대응 매크로 블록의 벡터 방향이 서로 일치하는지를 확인하였다. 비교를 위하여, 먼저 해당 매크로 블록에서의 움직임 벡터를 방향에 따라 여러 개의 bin값으로 양자화하고, 두 개의 부호화된 프레임에서의 각 대응 매크로 블록의 방향을 서로 비교하여 같은 bin값을 가질 경우 유효 움직임 벡터로 판단하였다. 이러한 유효 움직임 벡터 수를 이용하여 움직임 벡터 유효 비율을 얻을 수 있고 표 3에 이를 나타내었다. 그림 7은 유효 움직임 추정 벡터를 도시한 것이다. 그림 7-(c)는 제안된 알고리즘에서의 움직임 추정 벡터계를 보이고 있고, 그림 7-(d)는 Kobla 알고리즘에서의 움직임 추정 벡터계를 나타내고 있다. IPP 프레임의 움직임 벡터계를 보여주는 그림 7-(b)와 비교하였을 때, 제안된 움직임 재추정 방법이 Kobla의 방법보다 우수하다는 것을 잘 보여주고 있다. 그림 7에서 보이는 각 블록의 음영

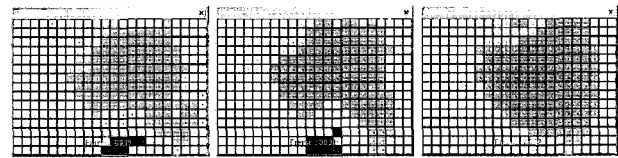
표 2. 움직임 추정 벡터의 수 비교

Table 2. Comparison of MDR.

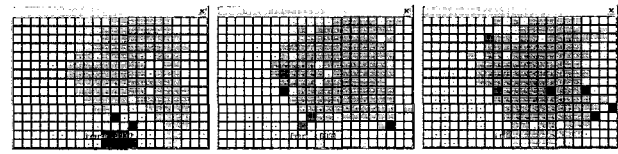
method frame	Proposed Method		Kobla's method [8]	
	Avg. MV Num.	MDR	Avg. MV Num.	MDR
R_i frame	198	0.71	176	0.63
B_1 frame	216	0.78	199	0.71
B_n frame	219	0.79	98	0.35
Total frame	212	0.76	163	0.58



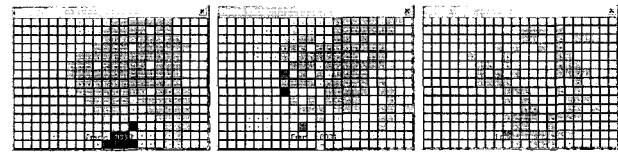
(a) 원 영상 (a) Original Image.



(b) IPP 부호화 프레임에서의 움직임 벡터 표현 (b) MV vector in IPP encoded frame.



(c) IPB 부호화 프레임에서의 움직임 추정 벡터 표현 (제안된 방법) (c) Estimated MV vector in IPB encoded frame (Proposed method).



(d) IPB 부호화 프레임에서의 움직임 추정 벡터 표현 (Kobla의 방법) (d) Estimated MV vector in IPB encoded frame (Kobla's method).

그림 7. 움직임 재추정 벡터의 예 (좌측부터 우측으로 : R_i , B_1 , B_n 프레임)

Fig. 7. Examples of flow estimation (From left to right : R_i , B_1 , B_n frames).

표 3. 유효 움직임 추정 벡터 비율 비교

Table 3. Comparison of the effective MDR.

method frame	Proposed Method	Kobla's method [8]
R_i frame	0.88	0.81
B_1 frame	0.75	0.60
B_n frame	0.82	0.43
Total frame	0.81	0.61

차는 움직임 추정 벡터의 방향을 지시하는 bin값을 표현한다. 그림 8은 IPB 부호화 시퀀스로부터의 움직임 벡터를 정규화하여 이를 IPP 부호화 프레임의 움직임 벡터와 그 방향에 따라 히스토그램 비교를 한 결과 그림이다. 움직임 벡터는 방향에 따라 세밀화될 수 있도록

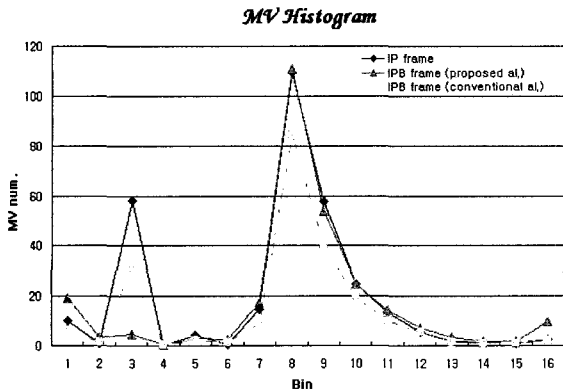


그림 8. 움직임 추정 벡터 히스토그램 비교
Fig. 8. Histogram comparison in N-MV field.

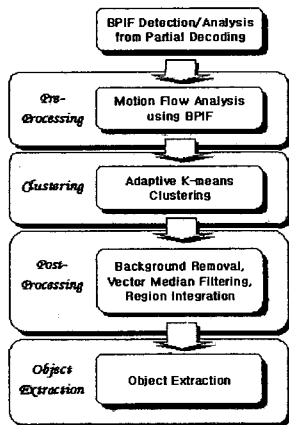


그림 9. 움직임 객체의 추적을 위한 전처리 과정
Fig. 9. Preprocessing stage for moving object tracking.

보다 많은 bin 을 가지게 하였다. 그림 7에서의 결과와 마찬가지로 제안된 방법에서의 움직임 추정 벡터 히스토그램이 Kobla의 방법보다 IPP 부호화된 프레임에서의 움직임 벡터 히스토그램에 더 근접한 것을 알 수 있다.

$$MV\ Accuracy\ Ratio = \frac{\text{Number of Effective MV in IPB frame}}{\text{Number of MV in IPP frame}}$$

V. 응용

본 장에서는 제안된 벡터계를 이용한 여러 응용분야에서의 결과를 보여준다. 증대된 정규화 벡터를 이용하여 움직임 객체를 갖는 프레임을 찾고, 움직임 객체를 분할하며, 해당 객체를 추적하는 등의 다양한 응용 분야를 기술하였다.

1. 정규화된 움직임 벡터 상에서의 객체 추출

본 논문에서는 벡터 평면에서의 움직임 객체의 분할

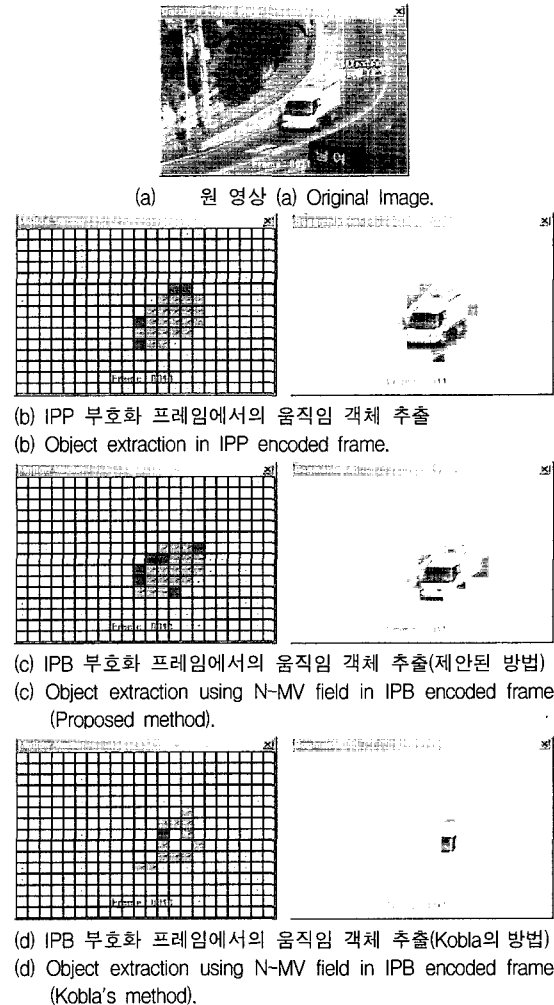


그림 10. 움직임 추정 벡터를 이용한 객체의 추출
Fig. 10. Object extraction using the normalized MV field.

을 위해 적응적인 K-means 클러스터링 알고리즘을 적용시켰다. 여기서의 클러스터링 알고리즘은 2차원 평면 상에서 움직임 벡터의 길이와 각을 이용하여 움직임 객체에 따른 영역 단위의 분할을 수행하게 된다. 그림 9 과 그림 10은 각각 본 논문에서 사용된 움직임 객체 추출 방법에 대한 블록도와 이를 이용한 객체 추출의 예를 도시하고 있다.

그림 10의 (b)~(d)에서 좌측 영상은 원 영상에 대한 정규화된 움직임 벡터계를 나타내고, 우측 영상은 좌측의 움직임 벡터계로부터 추출된 움직임 객체를 보여주고 있다. 보이는 바와 같이 제안된 알고리즘에서의 효율적인 움직임 재추정이 보다 정확한 움직임 객체의 추출에 많은 영향을 끼치는 것을 확인할 수 있다.

2. 정규화된 움직임 벡터 상에서의 객체 추출프레임 수의 비교

제안된 벡터 영역은 이를 이용한 객체 추적 알고리즘 등에도 유용하게 사용될 수 있는데 이는 움직임 재해석 과정으로부터 모든 프레임에서의 효율적인 움직임 해석이 가능해졌기 때문이다. 그림 9에서의 움직임 객체 추출 알고리즘을 제안된 벡터 추정 영역에 적용할 시, 표 4에서 보여지듯 기존의 벡터 추정 영역을 적용할 때와 비교하여 객체가 검출된 프레임의 비율이 평균 21% 이상 증가하는 것으로 확인되었다. 이는 각 프레임에서의 높은 유효 움직임 추정 벡터 수의 증가와 같은 의미를 나타내는 것이다.

3. 정규화된 움직임 벡터 상에서의 객체 추적

최종적으로 본 논문에서 이용하고 있는 정규화된 움직임 벡터계를 이용한 객체 추적 방법을 그림 11을 통해 보여주고 있다. 그림 11은 제안된 벡터계에서의 효과적인 움직임 객체 추적을 보여줌과 동시에 IPP시퀀스에서의 움직임 벡터를 이용한 객체 추적 기법과의 비교를 통해 제안하는 정규화 움직임 벡터 표현의 유효성 또한 증명하고 있다. 그림 11-(a)는 벡터 재추정 기법을 사용하지 않고 IPP시퀀스에서의 P프레임으로부터 순방향 움직임 벡터만을 이용한 객체 추적을 나타내고 있는데, 그림 11-(b)의 움직임 재추정을 이용한 객체 추적과 그 움직임 경로상에서 큰 차이를 보이지 않는 반면에 오히려 그림 11-(b)가 객체 추적에 있어 보다 더 유연한 경로를 갖는 것을 알 수 있다. 이는 움직임 벡터 재추정 기법이 실제 프레임 상의 움직임 벡터를 크게 변형시키거나 왜곡시키지 않는다는 것을 뜻하며, 또한 본 논문에서 제안하고 있는 벡터 재추정 기법의 유효성을 잘 보여주고 있다고 할 수 있다. 본 논문에서 사용된 움직임 객체 추적 기법은 [16]을 참고하였다.

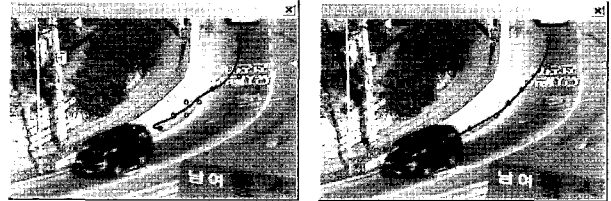
VI. 결 론

본 논문에서 우리는 MPEG 비디오에서의 움직임 벡터 정규화를 통한 재해석 알고리즘을 제안하였다. 실험 결과는 제안된 움직임 재추정 기법이 조밀하고, 포괄적인 움직임 벡터 표현이 가능하다는 것을 보여주고 있다. 또한, 실험 결과에서 보여주는 것과 같이 제안된 움직임 벡터 재해석 방법은 객체의 효과적인 추출 및 추적 기법 등 비디오에서의 다양한 움직임 해석을 가능케 한다. 영상에 대한 충분한 움직임 표현을 얻을 수 있는 제안된 움직임 벡터 재해석 기법은 비단 객체의 추적 기법에서뿐만 아니라 비디오 시퀀스에서의 전역적인

표 4. 객체 검출 프레임 비의 비교

Table 4. Comparison of the number of object detected frames.

method \ frame	Proposed Method	Kobla's method [8]
R_1 frame	0.59	0.47
B_1 frame	0.83	0.70
B_n frame	0.68	0.29
Total frame	0.70	0.49



(a) P프레임을 이용한 움직임 객체의 추적 (b) 움직임 재추정을 이용한 객체의 추적
(a) Object tracking in IPP encoded frame. (b) Object tracking in N-MV field.

그림 11. 움직임 객체의 추적

Fig. 11. Comparison of object tracking.

카메라 움직임 검출과 비디오 인덱싱이나 검색을 위한 움직임 서술자로서도 폭 넓은 활용이 가능 할 것이다. 그리고, 움직임 기반 비디오 인덱싱은 칼라나 질감과 같은 특징들에 기반한 인덱싱 기법과의 상관 조합으로써 유사도 기반의 비디오 검색 시스템의 성능을 크게 향상시킬 수도 있다.

참 고 문 헌

[1] H.L. Eng and K.K. Ma, "Bidirectional motion tracking for video indexing," Multimedia Signal Processing, pp. 153-158, 1999.
 [2] J.G. Kim, H.S. Chang, J.W. Kim and H.M. Kim, "Efficient camera motion characterization for MPEG video indexing," Proc. of IEEE International Conf. on Multimedia and Expo, vol. 2, pp. 1171-1174, 2000.
 [3] D. Zhong and S.F. Chang, "AMOS: an active system for MPEG-4 video object segmentation," Proc. of IEEE International Conf. on Image Processing, vol. 2, pp. 647-651, 1998.
 [4] M. Cherfaoui and C. Bertin, "Temporal segmentation of videos: A new approach," in IS & T and SPIE Proc.: Digital Video Compression Algorithm and Technology, vol. 2419, San Jose, 1995.
 [5] H. Zhang, A. Kankanhalli and S. Smoliar, "Automatic partitioning of full-motion video," Multimedia Systems I, 10-28, 1993.
 [6] H. Zhang, C.Y. Low, Y. Gong and S. Smoliar,

"Video parsing using compressed data," in Proc. of SPIE Conf. on Image and Video Processing II, vol. 2182, pp. 142-149, 1994.

[7] Y. Juan and S.F. Chang, "Scene change detection in a MPEG compressed video sequence," in IS & T and SPIE Proc.: Digital Video Compression Algorithm and Technology, vol. 2419, San Jose, 1995.

[8] V. Kobla, et al, "Indexing and Retrieval of MPEG Compressed Video," Journal of Electronic Imaging, Vol.7 (2), 1998.

[9] S.F. Chang, et al, "VideoQ: An automated content based video search system using visual cues," Proc. of ACM Multimedia, pp. 313-324, 1997.

[10] F. Liu and R.W. Picard, "Finding periodicity in space and time," Proc. of IEEE International Conf. on Computer Vision, pp. 376-383, 1998.

[11] C.W. Ngo, et al, "Motion characterization by temporal slice analysis," Proc. of IEEE International Conf. on Computer Vision and Pattern Recognition, pp. 768-773, 2000.

[12] E. Ardizzone, M. La Casica and D. Molinelli, "Motion and color-based video indexing and retrieval," Proc. of IEEE International Conf. on Computer Vision and Pattern Recognition, pp. 135-139, 1996.

[13] E. Ardizzone, et al, "Video indexing using MPEG motion compensation vectors," Multimedia Computing and Systems, pp. 725-729, 1999.

[14] ISO/IEC/JTC1/SC29/WG11, 13812-2, 1995.

[15] O. Sukmarg and K.R. Rao, "Fast object detection and segmentation in MPEG compressed domain," Proc. of IEEE TENCON 2000, vol. 3, pp. 364-368, 2000.

[16] N.W. Kim, T.Y. Kim and J.S. Choi, "Motion analysis using the normalization of motion vectors on MPEG compressed domain", ITC-CSCC2002, vol.3, pp.1408-1411, 2002.

— 저 자 소 개 —



김 낙 우(학생회원)
 1997년 2월 중앙대학교 제어계측 공학과 졸업(공학사).
 2002년 2월 중앙대학교 첨단영상대학원 영상공학과 졸업(공학석사).
 2002년 3월~현재 중앙대학교 첨단영상대학원 영상공학과 박사과정 재학중.

<주관심분야: 동영상부호화, 멀티미디어 데이터베이스, 영상통신>



최 중 수(정회원)
 1975년 2월 인하대학교 전기공학과 졸업(공학사).
 1977년 2월 서울대학교 전자공학과 졸업(공학석사).
 1981년 2월 일본 Keio University 전기공학과 졸업(공학박사).

1981년 9월~1999 8월 중앙대학교 전자공학과 교수.
 1999년 9월~현재 중앙대학교 첨단영상대학원 교수.
 <주관심분야: 컴퓨터비전, 영상정보기술>



김 태 용(정회원)
 1986년 2월 한양대학교 전기공학과 졸업(공학사).
 1998년 2월 한양대학교 전자통신공학과 졸업(공학석사).
 1998년 2월 포항공과대학교 컴퓨터공학과 졸업(공학박사).

2002년 1월~현재 중앙대학교 첨단영상대학원 교수.
 <주관심분야: 컴퓨터 비전, 컴퓨터 보안, 영상통신, 영상처리>