

論文2002-39SP-3-9

# 압축영역에서 움직임 벡터의 재추정을 이용한 비디오 해석 기법

## (Video analysis using re-constructing of motion vectors on MPEG compressed domain)

金 洛 禹 \* , 金 泰 鎔 \* , 姜 應 寬 \*\* , 崔 宗 秀 \*

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

### 요 약

본 논문은 MPEG 비디오에서 나타나는 여러 예측 형태의 움직임 벡터를 프레임 타입에 관계없이 단일 예측방향만을 갖도록 새롭게 추정하여 비디오 영상물의 분석에 직접적으로 활용하는 방법에 대해 제시하고 있다. 또한 재추정된 각 프레임에서의 움직임 벡터를 이용한 비디오 시퀀스 내에서의 객체 추출 및 추적 기법 등에 대해서도 새롭게 제안하였다. 제안된 알고리즘은 압축 영상에 대한 전체적인 복원과정을 거치지 않고, 압축 비디오 영역으로부터 쉽게 추출될 수 있는 매크로 블록 영역 상에서 수행되었으며, 실험 결과는 제안된 방법의 높은 성능을 잘 나타내어 주고 있다.

### Abstract

A macroblock(MB) in MPEG coded domain can have zero, one, or two motion vectors depending on its frame type and prediction direction (forward-, backward-, or bi-directionally). In this paper, we propose a method that converts these motion vectors on MPEG coded domain as a uniform set, independent of the frame type and the direction of prediction, and directly utilizes these re-analyzed motion vectors for understanding video contents. Also, using this frame-type-independent motion vector, we propose novel methods for detecting and tracking moving objects with frame-based detection accuracy on the compressed domain. These algorithms are performed directly from the MPEG bitstreams after VLC decoding with little time consumption. Experimental results show validity and outstanding performance of our methods.

**Keyword** : video parsing, motion analysis, object extracting and tracking

\* 正會員, 中央大學校 尖端映像大學院 映像工學科  
(Dep. of Image Engineering, Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University)

\*\* 正會員, 全州大學校 情報技術컴퓨터工學部 멀티미디어工學科

(Dep. of Multimedia Engineering, Div. of Computer Science and Engineering, Jeon-Ju University)

※ 본 연구는 교육부의 두뇌한국 21 사업(BK21)과 과학기부의 국가지정연구실(2000-N-NL-01-C-285) 그

리고 정보통신연구진흥원의 연구지원으로 수행되었습니다.

接受日字:2001年12月10日, 수정완료일:2002年4月15日

### I. 서 론

최근 디지털 비디오 정보의 폭발적 증가에 따라 효과적으로 영상을 관리하기 위한 비디오 데이터베이스 관리 시스템(Video Database Management System)이나 내용 기반 비디오 표현 기술 등에 관련된 연구들이

폭 넓게 진행되고 있다. 이에, 보다 효과적으로 비디오 데이터를 분석하고 이해하기 위한 비디오 파싱(parsing) 기술, 파싱된 데이터에의 신속한 접근과 이용을 위한 비디오 인덱싱(indexing) 기술 등이 점차적으로 크게 요구되고 있다.

일반적으로 비디오 파싱을 통해 분할된 샷(shot)의 내용은 크게 샷이 표현하고자 하는 기본적인 장면 요소와 카메라나 객체의 움직임 요소로 구분된다. 샷을 기반으로 하는 비디오 분석(analysis)방법은 결국 이 두 가지 요소를 얼마만큼이나 효과적으로, 그리고 자동적으로 비디오 데이터로부터 추출하여, 내용기반 검색을 위한 비디오 인덱싱 구조에 활용되게 할 수 있는가에 그 초점이 맞춰져 있다<sup>[1-2]</sup>.

움직임(motion)은 샷에서의 이러한 두 가지 요소, 즉, 기본적인 장면요소와 카메라의 움직임 요소 모두에 대해 매우 중요한 성분이기 때문에, 지능적인 비디오 브라우징/분석 등의 목적을 위해 샷의 움직임을 특징짓고 그 움직임을 분할, 추적하는 방법 등이 최근 활발히 연구되고 있는 상황이다<sup>[3-4]</sup>.

본 연구에서는, MPEG 압축 영역에서의 움직임 벡터 정보를 이용하여 비디오에서의 효율적인 움직임 해석 방법을 제안하고 있다. MPEG시퀀스로부터 추출된 움직임 벡터의 효과적인 재해석을 통하여, 영상 내 객체의 좌표 및 움직임 각도 등을 고려하던 기존의 알고리즘<sup>[5-7]</sup>과는 달리 움직이는 객체의 속도와 가속도 등을 고려한 새로운 객체 추출 및 추적 알고리즘을 제시하고 그 성능을 비교하였다.

제안된 알고리즘을 통해 얻어진 움직임 추정 벡터는 특히 비디오 시퀀스에서의 전역적인 카메라 움직임 검출과 비디오 인덱싱, 검색을 위한 특징 정보로서도 그 활용이 가능하다. 우선, 카메라의 전역적인 움직임 정보만으로 검출이 가능한 패닝(panning)이나 줌(zooming)등의 움직임 영역의 추출은 제안된 움직임 재추정 기법을 이용한 벡터 영역에서 좀 더 정확하고 정밀한 움직임 구간의 검출이 이루어 질 수 있다. 그리고, 비디오 인덱싱을 위한 특징 정보의 활용 측면에서도 MPEG 압축 영역에서의 움직임 벡터 정보만을 이용하는 본 알고리즘의 특성상, 영상의 색상이나 에지 정보 등을 고려하는 다른 인덱싱 알고리즘과는 달리 움직임 벡터를 인덱싱 정보로 활용함으로써 영상의 완전한 디코딩(decoding)에 따른 시간적인 손실과 인덱싱 데이터를 위한 영상 저장 공간 등을 크게 줄일 수 있다. 또한,

인덱싱 데이터의 검색 시에도 벡터 공간에서의 적은 데이터량으로 인해 영상의 검색 시간을 단축시킬 수도 있다.

본 논문의 순서는 다음과 같다. 우선, 2장에서는 MPEG 비디오 시퀀스의 프레임 구조에 대한 개략적인 설명이 이루어지고, 3장에서는 MPEG에서의 움직임 해석을 위한 기존의 알고리즘과 제안된 알고리즘을 서로 비교· 분석한다. 4장에서는 3장에서 제안된 움직임 재추정 알고리즘을 이용한 새로운 객체 추출 기법을 제시하고, 마지막으로, 5장과 6장에서 이에 대한 실험 결과 및 결론을 도출한다. 그림 1은 본 연구의 대략적인 블록도를 나타낸 것이다.

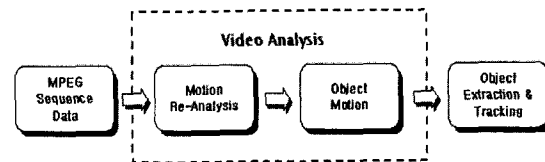


그림 1. 제안된 알고리즘의 블록도  
Fig. 1. Block diagram of the proposed algorithm.

## II. MPEG 시퀀스의 기본 구조

MPEG-1스트림은 일반적으로 그림 2와 같이 I 프레임, B 프레임, P프레임 등의 세 가지의 유형을 갖는 GOP(group of pictures)의 시퀀스로 구성되어 있다. I 프레임은 인트라(intra) 부호화 영상으로서 프레임간 예측을 쓰지 않고 생성되며, JPEG 알고리즘 등에서도 사용된다. 이 프레임은 GOP의 독립성을 유지하기 위해 필요로 된다. P 프레임은 예측 부호화 영상으로서 이전 I 프레임으로부터의 대응 매크로 블록을 이용하여, 순방향 움직임 예측을 통해 압축된다. P 프레임내의 매크로 블록 타입은 인트라 부호화 블록과 인터(inter) 부호화 블록 양쪽을 모두 포함하고 있다. B프레임은 쌍방향 예측 부호화 영상의 약칭으로서 앞뒤로 가장 인접된 레퍼런스 프레임(I프레임이나 P프레임)으로부터 쌍방향 움직임 예측을 통해 얻어진다. B 프레임 내의 매크로 블록 타입은 인트라 부호화 블록과 순방향 인터 부호화 블록, 역방향 인터 부호화 블록과 쌍방향 인터 부호화 블록의 네 가지 타입을 가질 수 있다. 결과적으로 P 프레임은 하나의 레퍼런스 영상으로부터 예측 부호화 되기 때문에, 매크로 블록이 가질 수 있는 최대의 움직임 벡터의 수는 하나가 되지만, B 프레임의 경우, 쌍방

향 영상으로부터 예측 부호화되기 때문에 하나의 매크로 블록이 두 개의 움직임 벡터를 가질 수도 있다<sup>[8]</sup>.

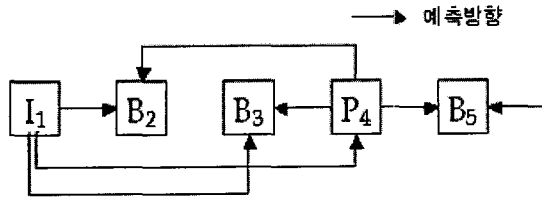


그림 2. MPEG 스트림에서의 일반적인 GOP 구조  
Fig. 2. General GOP structure in MPEG stream.

즉, 압축 스트림에서의 매크로 블록은 같은 프레임 내에서조차 그 프레임의 유형에 따라 다른 수의 움직임 벡터를 가지게 되고 이러한 움직임 벡터는 예측 방향 또한 일률적이지 않게 된다. 이는 그 매크로 블록이 인트라 부호화 블록인지, 순방향이나 역방향 혹은 양방향 예측 블록인지에 따라 달라진다. 따라서, 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 일관성 있는 움직임 벡터로 전환하는 작업이 먼저 선행되어야 한다.

### III. 움직임 벡터의 재추정을 이용한 움직임 해석

2장에서 제시된 바와 같이 움직임 벡터를 이용하여 프레임 내의 움직임을 추정하고자 할 때에는 일반적으로 프레임 유형에 상관없이 일관된 형태의 벡터 유형만을 갖도록 각각의 프레임을 먼저 재구성하게 된다. 이 논문에서는, 현재 프레임에서의 각각의 움직임 벡터를 다음 프레임 대응 블록에서의 역방향 벡터로 일괄적으로 표현함으로써 전체의 프레임 시퀀스를 역방향 벡터만을 가지는 프레임들로 재구성하였다.

#### 3.1 일반적인 벡터 추정 방법

MPEG 시퀀스의 GOP구조 상에서 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 B프레임의 수)라고 한다면, 이러한 프레임들 간의 유기적 관계를 통해 전체 시퀀스의 움직임을 역방향 움직임 벡터만을 갖는 프레임 시퀀스로 재 유추할 수 있다

<sup>[9-10]</sup>. 그 방법을 크게 세 단계로 나눌 때, 각각  $R_i$ ,  $B_n$  프레임에서의 움직임 재추정 기법과  $B_1$ 부터  $B_{n-1}$  프레임에서의 움직임 추정 방법으로 분류할 수 있는데, 본 논문에서는  $B_1$ 부터  $B_{n-1}$  프레임에서의 움직임 추정 방법으로 기존의 방법과 동일한 알고리즘<sup>[10]</sup>을 적용하였기 때문에  $R_i$ ,  $B_n$  프레임에서의 움직임 재추정 기법만을 기존의 방법과 비교하기로 한다.

첫 단계인  $R_i$ 프레임에서의 움직임 벡터 추정은 참조 프레임  $R_i$ 와  $B_1$ 프레임 간의 순방향 예측된 움직임 벡터를 이용한다. 만약, 매크로 블록  $(B_1)_{u,v}$  ( $u,v$ 는 해당 프레임의 매크로 블록들 중에서 현재 매크로 블록의 위치를 표시)에서의 움직임 벡터가 블록 크기 이하의 움직임을 갖는다고 가정할 때,  $B_1$ 프레임에서의 매크로 블록  $(B_1)_{u,v}$ 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 있다. 역으로 생각해 볼 때  $R_i$ 프레임에서의 매크로 블록  $(R_i)_{u,v}$ 는  $B_1$ 프레임에서의 매크로 블록  $(B_1)_{u,v}$ 에 대응하는 움직임 벡터  $(-x,-y)$ 를 갖는다고 유추해 볼 수 있고, 이로부터 우리는 참조 프레임  $R_i$ 에서의 움직임 추정 벡터를 얻는다. 매크로 블록  $(B_1)_{u,v}$ 가 순방향 벡터를 가지지 않은 경우의 매크로 블록  $(R_i)_{u,v}$ 의 움직임 추정 벡터는,  $B_1$ 프레임 이후로 연속된 B프레임들 중 순방향 벡터를 가진 매크로 블록  $(B_k)_{u,v}$ 의 움직임 벡터를 프레임 간격  $k$ 로 나눈 값으로부터 얻어질 수 있다. 매크로 블록  $(B_1)_{u,v}$ 에서의 움직임 벡터 길이가 블록 크기 이상일 경우 이에 대응하는  $R_i$ 프레임에서의 매크로 블록이  $(R_i)_{u,v}$ 라고 생각하는 어려우나 본 알고리즘이 모든 프레임 시퀀스에서의 움직임을 재유추하고 있으므로, 한 프레임이 통상 영상에서의 1/30초를 대표한다고 할 때, 각 프레임의 매크로 블록에서 벡터 길이가 블록 크기 이상일 확률은 매우 적다.

또한, 매크로 블록  $(B_1)_{u,v}$ 에서의 벡터 길이가 블록 크기를 벗어날 경우에도 이에 대응 하는 매크로 블록은  $u,v$ 의 인접 영역  $-1 \leq u,v \leq 1$  을 벗어나지 않으므로, 전체적인 관점에서 프레임의 움직임을 파악하는 데에는 크게 문제가 되지 않는다<sup>[10]</sup>. 벡터 추정의 두 번째 단계는 참조 프레임  $R_i$ 와  $B_n$ 프레임 간의 역방향 예측된 움직임 벡터를 이용한  $B_n$ 프레임에서의 움직임 벡터 추정이다. 벡터의 추정 방법은 첫 단계에서의 경우와 거의 유사하나, 위의 경우와는 달리  $R_i$ 프레임에 대한  $B_n$  프레임에서의 벡터는 역방향 움직임 벡터이므로 벡터

를 다시 역으로 변환할 필요는 없다.

그림 3은 이에 대한 설명을 나타낸 것이다. 실선은 실제적인 매크로 블록의 움직임 벡터 방향을 표시한 것이고, 점선은 이를 통해 재추정된 움직임 벡터의 방향을 표시하고 있다. 기존의 알고리즘에서는 이와 같은 방법으로  $R_i, B_n$  프레임에서의 움직임을 재추정 한다.

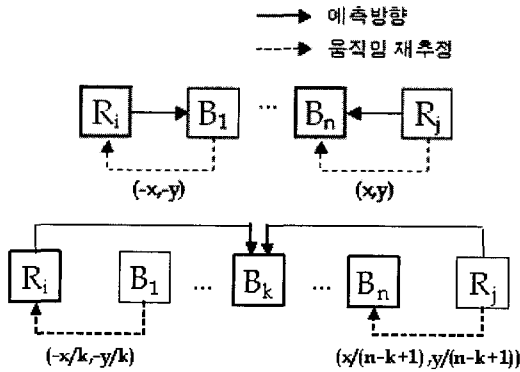


그림 3. 기존 방법을 이용한  $R_i, B_n$  프레임에서의 벡터 재추정 방법  
 Fig. 3. Re-construction of motion vectors by the conventional method in  $R_i, B_n$  frames.

그러나, GOP 구조에서의 이러한 벡터 추정 방식은 각각의 프레임을 나타내기엔 충분할 만큼의 유효 움직임 벡터를 추출하지 못하다는 문제점을 가지고 있다. 위의 서술에서 이미 나타난 바와 같이,  $R_i$  프레임에서의 움직임 벡터 추정은 B 프레임 상에서의 순방향 움직임 벡터를 이용하고,  $B_n$  프레임에서의 움직임 벡터 추정은  $R_j$  프레임으로부터 역방향 예측된 B 프레임 상에서의 역방향 움직임 벡터만을 고려하고 있다. 이는 MPEG 시퀀스에서의 프레임 구조가 쌍방향 예측이 가능하다는 것을 간과한 것인데,  $B_n$  프레임이 비단  $R_i$  프레임에서의 역방향 움직임 벡터 뿐만 아니라  $R_j$  프레임으로부터 순방향 움직임 벡터 또한 가질 수 있다는 점, 그리고  $R_i$  프레임에서의 벡터 추정 시 또한 위와 마찬가지로  $B_1$  프레임이 순방향 움직임 벡터 만이 아닌  $R_j$  프레임으로부터의 역방향 움직임 벡터를 가질 수도 있다는 점 등이 이전 방법에서는 고려되지 않고 있다.

3.2 제안된 벡터 추정 방법

이러한 문제점을 보정하기 위하여 제안된 방법에서는  $R_i$  프레임으로부터  $R_j$  프레임으로의 순방향 벡터로부터  $B_1$  프레임과  $B_n$  프레임에서의 벡터 예측을 보조함으로써 어느 한 방향에서의 벡터 예측이 아닌 쌍방

향 예측을 가능하게 한다. 즉,  $R_i$  프레임 상의 벡터 추정을 위해  $B_1$  프레임에서의 순방향 벡터만을 사용한 기존의 방법과 달리  $R_j$  프레임에서  $B_1$  프레임으로의 역방향 벡터와  $R_i$  프레임에서  $R_j$  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 이용한다. 단,  $R_j$  프레임이 I 프레임일 경우  $R_i$ 에서  $R_j$ 로의 순방향 벡터를 찾을 수 없으므로 이러한 방법을 이용할 수 없다.

이 알고리즘의 수식적 표현은 수식 1과 같다.  $B_n$  프레임 상에서 현재 매크로블록의 순방향 움직임 벡터를  $\overline{B_n R_i}$ 로 하고, 두 레퍼런스 프레임  $R_i, R_j$  간 각 매크로블록의 순방향 움직임 벡터를  $\overline{R_i R_j}$ 로 할 때,  $R_i$  프레임에서  $B_n$  프레임으로의 역방향 추정 벡터  $\overline{B_n R_j}$ 은 다음의 수식으로 쉽게 얻을 수 있다.

$$\overline{B_n R_j} = \overline{B_n R_i} - \overline{R_i R_j} \tag{1}$$

$\overline{B_n R_j}$ 를 얻기 위해  $R_j$ 에서  $B_n$ 으로의 역방향 벡터 성분만을 고려하던 그림 3의 기존 방법과는 달리 제안된 방법에서는 순방향 벡터 성분  $\overline{B_n R_i}, \overline{R_i R_j}$  등도 이용함으로써 그림 4에서와 같이 보다 효과적인 벡터 추정을 가능하게 하였다.

또한,  $B_1$  프레임 상에서 현재 매크로블록의 역방향 움

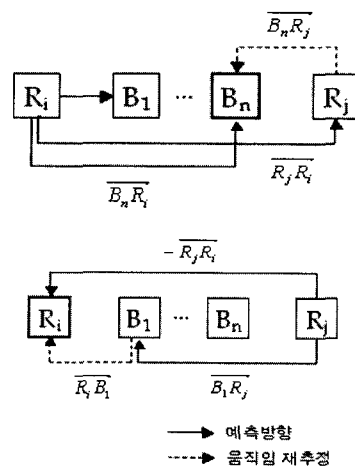
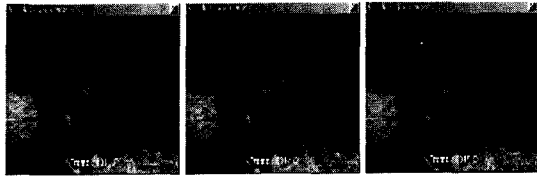
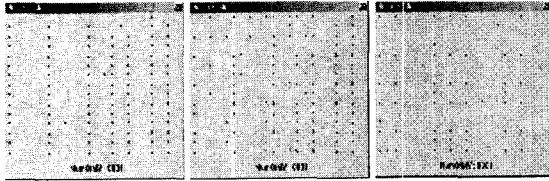


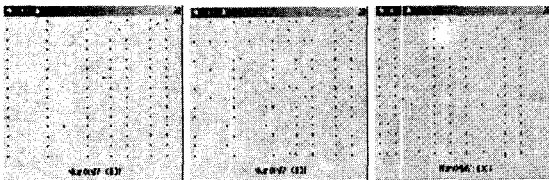
그림 4. 제안된 알고리즘을 이용한  $R_i, B_n$  프레임에서의 벡터 재추정 방법  
 Fig. 4. Re-construction of motion vectors by the proposed method in  $R_i, B_n$  frames.



(a) 원 영상  
(a) Original sequences.



(b) 기존의 알고리즘에 의한 움직임 재추정  
(b) Motion flow representation by the conventional method.



(c) 제안된 알고리즘에 의한 움직임 재추정  
(c) Motion flow representation by the proposed method.

그림 5. 움직임 벡터 재추정(좌측부터 우측으로 각각  $R_i$ ,  $B_i$ ,  $B_2$  프레임)

Fig. 5. Motion flow analysis( $R_i$ ,  $B_i$  and  $B_2$  frames from left-side to right-side).

직임 벡터를  $\overline{B_i R_i}$ 로 했을 때,  $B_i$  프레임에서  $R_i$  프레임으로의 역방향 추정 벡터  $\overline{R_i B_i}$ 은 다음과 같이 얻을 수 있다.

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

이와 같은 방법으로 기존의 알고리즘을 보완할 경우 그림 5-(c)에서 보여지는 바와 같이  $B_i$ 나  $B_2$  프레임에서의 예측 벡터의 수를 그림 5-(b)의 기존의 알고리즘에 비해 크게 늘릴 수 있고, 이를 통해 프레임에서의 효과적인 특징 정보로서 이러한 예측 벡터들을 사용할 수 있게 된다.

#### IV. 압축 영역 상에서의 움직임 검출

본 장에서는 3장에서 제안된 방법을 통해 추출되어진 각 프레임에서의 움직임 추정 벡터를 이용하여 객체의 움직임을 가지는 연속적인 프레임에서의 객체의 분할 및 추적에 대한 알고리즘을 제안한다. 추출된 움직임 벡터 공간에서의 노이즈 제거를 위해 벡터 미디언 필터링 과정을 먼저 수행한다.

##### 4.1 움직임이 있는 객체의 추출

움직임 객체의 추적을 위해서는 먼저 프레임으로부터 객체를 추출해내는 작업이 선행되어야 한다. 본 논문에서는 벡터 평면에서의 움직임 객체의 분할을 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. 이러한 영역 분할의 결과로서 영상에서의 배경과 객체가 각각의 영역으로 분할되게 되는데, 정확한 객체의 추출과 추적을 위해서 우선 배경 영역을 제거한다. 배경 영역은 대상 영상과 분할된 영역의 가로-세로 크기를 서로 비교, 배경 영역과 유사한 크기를 갖는 분할 영역을 찾음으로써 손쉽게 제거가 가능하다. 그리고 나서, 데이터 취득 과정의 오류나 환경적 요인 등으로 인해 발생한 노이즈를 제거하기 위해 배경이 제거된 이미지 상에 벡터 미디언 필터링을 적용시킴으로써 전체 벡터 영역을 스무딩(smoothing) 한다. 이러한 과정을 거쳐 얻어진 영역이 일정한 크기 이상일 때, 최종적인 객체 영역으로 분리된다. 그러나, 만일 영역이 크기가 정해진 값보다 작을 경우 아래의 수식을 사용하여 현재 객체 영역( $x$ )과 인접 영역들( $k$ )과의 유사도 비교를 통해 현재 영역은 가장 가까운 특징을 가지는 인접 영역에 병합된다.

$$\min_{0 < k < T} \left( \sqrt{(L(x) - L(k))^2 + w^2 (A(x) - A(k))^2} \right) < Thre\_R \quad (3)$$

여기서  $L(x)$ 는 현재 객체 영역  $x$ 내의 벡터 평균 길이이고,  $L(k)$ 는  $k$ 번째 객체 영역내의 벡터 평균 길이를 나타낸다.  $A(x)$ 와  $A(k)$ 는 각각 현재 영역과  $k$ 번째 영역에서의 벡터 평균 각도이다. 그리고,  $w$ 는 벡터 길이와 각도에 대한 수치계산 값의 보정을 위한 가중치 이

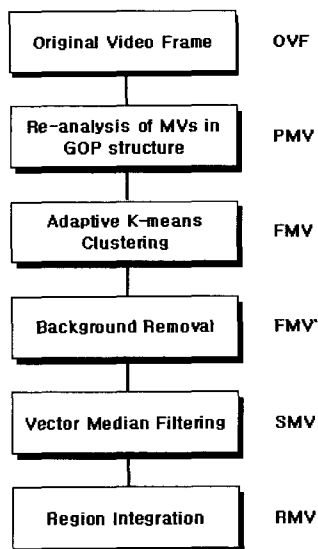
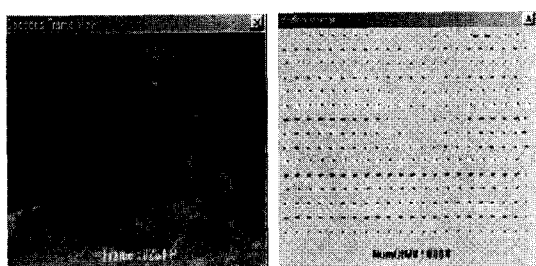
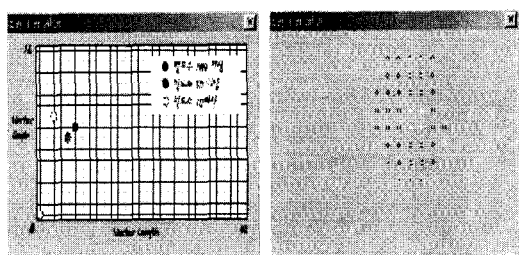


그림 6. 움직임 객체의 추적을 위한 전처리 과정  
 Fig. 6. Preprocessing stage for moving object tracking.



(a) 원 영상 (b) 움직임 벡터  
 (a) Original image. (b) Motion vectors.



(c) L- $\theta$  공간에서의 누적 빈도수 (d) 움직이는 객체의 검출  
 (c) Cumulative frequency in L- $\theta$  domain. (d) Detection of moving object.

그림 7. MPEG 영상에서 움직임 벡터를 이용한 객체의 추출  
 Fig. 7. Object extraction using motion vectors in MPEG sequences.

고,  $T$ 는 현재 프레임에서의 최대 영역 개수이다. 이로부터 우리는 최종적으로 프레임에서의 객체 영역

을 얻을 수 있게 된다. 그림 6과 그림 7은 각각 본 논문에서 사용된 움직임 객체 추출 방법에 대한 블록도와 이를 이용한 객체 추출의 예를 도시하고 있다.

4.2 움직이는 객체의 추적

움직이는 객체의 추적은 영상 시퀀스에서의 시공간적 관계에 의해서 수행된다. 임의의 시퀀스에서 동일 객체의 움직임은 짧은 시간 간격에서 선형성을 갖는 움직임으로 근사화 될 수 있기 때문에, 현재의 프레임에서의 움직임 특징은 일반적으로 시간적으로 인접한 프레임에서의 움직임 특징과 비슷하다. 이러한 특징을 이용하여 현재 프레임에서 추출된 영역을 이전 프레임에서의 영역과 비교, 링크시킴으로써 시퀀스에서의 전체적인 움직임 궤적을 추적할 수 있다.

움직임 객체의 추적 알고리즘은 크게 두 부분으로 나누어 진다. 첫 단계는, 두 프레임 A, B에서 그림 6의 과정을 통해 추출된 각 영역들 간의 공간상의 중심좌표를 이용, 일차적으로 중심점간 최소 거리 계산을 통해 프레임간 대응 영역을 찾는다. 그림 8-(a)와 같이 이전 프레임  $i-1$ 에서의 영역  $x$ 의 중심좌표를 현재 프레임  $i$ 에서의 영역 A, B, C의 중심좌표와 비교하여 이 중 최단 거리를 가진 영역과 일차적으로 서로 대응시킨다. 그러나, 이렇게 찾아진 대응 영역들은 단지 거리상의 근접성만을 고려한 것이기 때문에 이러한 영역들이 서로 같은 객체인가에 대한 좀 더 정확한 탐구가 두 번째 단계에서 요구된다. 이 논문에서는 현재의 영역에 대응된다고 생각되는 이전 영역으로부터 영역의 최대 속도와 가속도를 유추하여 대응점  $C_p$ 를 구하고 이 대응점과 이전 영역에서의 영역 중심점  $C_{i-1}$ 과의 길이 및 현재 영역에서의 영역 중심점  $C_i$ 과  $C_{i-1}$ 과의 길이를 서로 비교함으로써 영역의 연속성을 검증하였다. 그림 8-(b)와 8-(c)는 각각 이전 영역에서의 객체 움직임을 고려한  $C_p$ 점 탐색 방법과 수식 (4)에서  $C_p, C_i, C_{i-1}$ 좌표의 공간적 관계에 대한 표현을 도시하고 있다. 영상에서의 객체의 움직임이 일정한 선형성을 갖는 것을 고려하였을 때, 식 (5)에서 좌표  $C_p$ 는 좌표  $C_{i-1}$ 과 서로 연관되므로  $C_i$ 의 좌표가 급격히 변하지 않는 한, 식 (4)의 수식은 항상 만족하게 된다. 또한 좌표  $C_i$ 와  $C_{i-1}$ 이 크게 다르더라도 좌표  $C_p$ 가 이전 영역으로부터의 속도 값과 가속도 값에 대한 위치 정보를 가지고 있으므로 선형성을 갖는 급격한 이동에 대해서도 식 (4)를 만족하게 된다. 우리는 이와 같은 두 단계를 거쳐 이전 영역과 현재 영역과의 객체의 연속성을 파악하였다. 여기에서,

두 영역에서의 중심 각도의 흐름과 움직임 벡터의 수의 유사성도 역시 고려되어야 할 사항이다. 결과적으로, 현재의 영역이 아래 수식들을 만족할 때, 이전 프레임의 영역과 비로서 서로 링크되고 우리는 전(全) 영상 시퀀스에 대한 객체의 움직임을 추적할 수 있게 된다.

$$|C_{i-1} - C_i| < |C_p - C_{i-1}|, \tag{4}$$

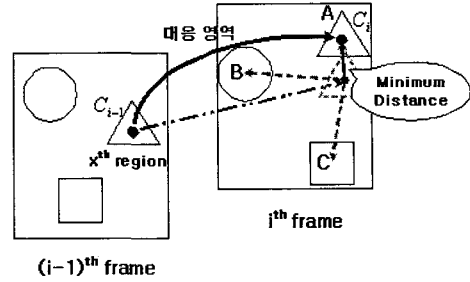
$$C_p = C_{i-1} + \beta \cdot (\bar{v} + \bar{a}) + \gamma, \tag{5}$$

$$\bar{v} = \frac{1}{k} \sum_{t=1}^k |C_{i-t} - C_{i-t-1}|, \quad \bar{a} = \frac{1}{k} \sum_{t=1}^k |v_{i-t} - v_{i-t-1}|,$$

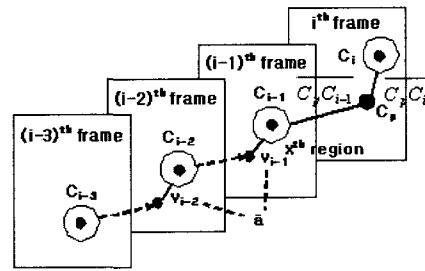
여기서,  $C_p$ 는  $i-1$ 프레임에서의 영역  $x$ 에 대응하는 이전 영역으로부터 속도, 가속도 값을 유추하여,  $i-1$ 프레임에서  $i$ 프레임으로의 영역  $x$ 의 움직임을 예측하는 좌표값을 나타낸다. 두 프레임  $i-t$ 와  $i-t-1$  간의 영역 이동 속도  $v_{i-t}$ 는 프레임 간 영역 중심점 차분  $|C_{i-t} - C_{i-t-1}|$ 이며,  $\bar{v}$ 와  $\bar{a}$ 는 각각 정수의 값을 갖는 이전 영역의 평균 속도와 평균 가속도를 나타낸다.  $\beta$ 는 비례 정수이고, 객체의 움직임이 선형적이라고 가정할 때 통상적으로 2의 값을 갖게 된다.  $\gamma$ 는 오프셋으로  $\bar{v} + \bar{a}$ 의 값이 0일 때는 2의 값을 갖고, 그 외일 때는 0의 값을 갖는다. 연속적인 움직임 영역에서의  $\bar{v} + \bar{a}$ 값은 프레임 간 짧은 시간 간격으로 인해 일반적으로 0의 값을 갖게 되는데, 이 때에  $C_p$ 값에 대한 최소 예측 범위로써  $\gamma$ 의 오프셋값을 이용한다.  $\bar{v} + \bar{a}$ 값이 0이 아닐 경우는 프레임 간 객체의 움직임이 비교적 큰 경우인데, 이 때에는 비례 정수  $\beta$ 값을 이용,  $C_p$ 값에 대한 예측 범위를 넓혀 주게 된다.

이미 전송된 바와 같이 샷-기반 비디오 인덱싱 방법에서 객체의 움직임은 샷 단위나 서브-샷(sub-shot) 단위로 분할된 비디오 시퀀스의 내용을 간략하게 기술해주는 효과적인 특징요소라 할 수 있다. 이는 샷이라는 비디오에서의 단위는 결국 다양한 객체의 움직임을 가지는 일련의 프레임을 나타내는 것이기 때문이다.

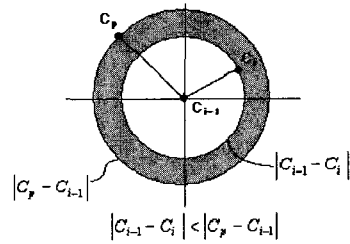
본 장에서 제안한 알고리즘의 성능은 본질적으로 각각의 프레임을 묘사하는 움직임 벡터의 정확성과 유효 벡터의 수 등에 매우 민감한 영향을 받게 된다. 결국 3장에서 제안된 각 프레임에서의 효과적인 벡터 추정 방법이 본 장에서 제안된 알고리즘의 성능 향상에 큰 영향을 미친다.



(a) 프레임간 대응 영역의 탐색  
(a) Search of corresponding region between two frames.



(b)  $C_p$  탐색  
(b)  $C_p$  search.



(c) 대응 조건  
(c) Corresponding condition.

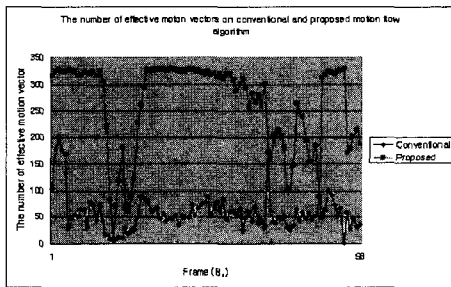
그림 8. 이전 영역과의 움직임 관계를 고려한 객체 추적  
Fig. 8. Object tracking based on the motion relation among previous regions.

### V. 실험 결과

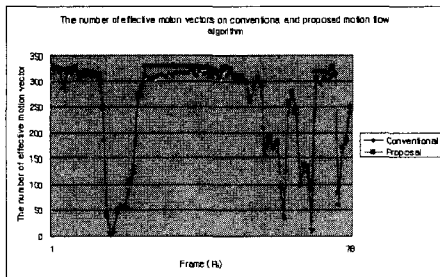
이 장에서는 3장에서 제안된 움직임 벡터 추정 기법에 대한 유효성 검토와 4장에서 제안된 객체의 움직임 추출 알고리즘에 대한 성능을 제시한다. 본 논문에서는 다양한 카메라의 움직임과 객체의 움직임을 가지고 있는 MPEG-1 스트림으로부터 제안된 알고리즘의 성능을 평가하였다.

그림 9는 기존의 움직임 추정 방법과 제안된 방법에서의 유효벡터 수를 비교하여 보여주고 있다.  $B_n$  프레

입에서의 추정 움직임 벡터는 제안된 방법에서 3배 이상의 수적 증가를 보여주고 있는데, 이러한 경향은 실험 영상의  $B_n$  프레임이 역방향의 벡터 성분을 많이 가지지 않을수록 더욱 두드러지게 나타난다. 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 것이다.



(a)  $B_n$  프레임에서의 움직임 추정 벡터의 수  
(a) The number of normalized motion vectors in  $B_n$ .



(b)  $R_i$  프레임에서의 움직임 추정 벡터의 수  
(b) The number of normalized motion vectors in  $R_i$ .

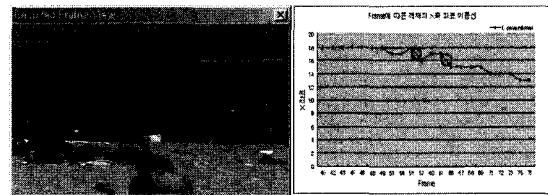
그림 9. 유효 움직임 추정 벡터 수 비교  
Fig. 9. Comparison of the number of normalized motion vectors.

$R_i$  프레임에서는  $B_n$  프레임 보다 상대적으로 적은 10% 정도의 유효벡터 성분의 증가가 나타나는데 이는 일반적인  $B_n$  프레임 상에서 움직임 벡터의 대부분은 순방향 벡터이기 때문이다. 통상적으로  $n=2$ 의 값을 가지므로 전체 시퀀스 프레임의 2/3 정도가 제안된 알고리즘에 의해 개선된 벡터 영역을 갖게 된다.

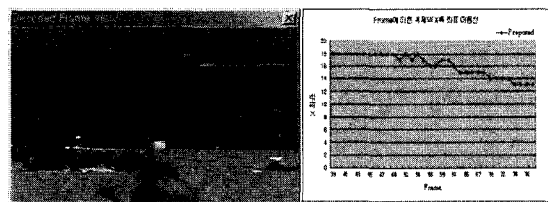
또 제안된 벡터 영역은 이를 이용한 객체 추적 알고리즘 등에도 유용하게 사용될 수 있는데 이는 움직임 재해석 과정으로부터 모든 프레임에서의 효율적인 움직임 해석이 가능해졌기 때문이다. 4.1절의 움직임 객체

표 1. 객체 검출 프레임 수 비교  
Table 1. Comparison of the number of object detected frames.

Sequence	Original Resolution	The number of total frame	Detectable frame (conventional)	Detectable frame (proposed)	Improved rate (%)
Boat	336x208	83	18	28	35
Truck	352x240	238	54	120	55
Van	352x240	130	44	116	62



(a) 일반적인 벡터 추정 방법을 이용한 움직임 객체의 추적  
(a) Tracking of moving object with a conventional motion analysis method.



(b) 제안된 벡터 추정 방법을 이용한 움직임 객체의 추적  
(b) Tracking of moving object with a proposed motion analysis method.

그림 10. 객체 추적 결과  
Fig. 10. Results of object tracking.

추출 알고리즘을 제안된 벡터 추정 영역에 적용할 시, 표 1에서 보여지듯 일반적인 벡터 추정 영역을 적용할 때와 비교하여 객체가 검출된 프레임의 비율이 평균 50% 이상 증가하는 것으로 확인되었다. 이는  $B_n$  프레임에서의 높은 유효 추정 벡터 수의 증가와 같은 의미를 나타내는 것으로  $B_n$  프레임에서의 객체 추출 비율 상승에 의한 효과와 레퍼런스 프레임  $R_i$ 를 이용하여 추출된 벡터 수의 증가에 의한 것이다.

MPEG 시퀀스에서의 이러한 객체 검출 프레임 수의 증가는 그림 10에 나타난 바와 같이 움직임 객체의 추적에도 그 영향을 미친다. 4.2절에서 제안된 객체 추적 알고리즘을 동일 적용하였을 때, 제안된 벡터 추정 영역 하에서의 움직임 객체의 추적은 그림 10에서 보이는



바와 같이 첫 프레임부터 끝 프레임까지 동일한 객체로 인식하고 올바르게 추적하였으나, 일반적인 벡터 추정 영역에서의 움직임 객체의 추적에서는 두 번의 객체 추적 실패를 보이고 있다.

이러한 이유는 50% 이상 줄어든 객체 검출 프레임수로 인해 생기는 객체 중심 영역의 불연속성에 기인한다. 즉, 중간 프레임의 공백으로 인해 생기는 객체 중심 좌표의 급격한 움직임이 비선형적 특성을 나타내기 때문이다.

그리고, 시퀀스에서의 객체 영역은 일반적으로 영역 내 움직임 벡터 분포의 가변으로 인하여 프레임마다 그 중심 좌표가 매크로 블록 단위로 일정한 유동을 갖게 된다. 때문에, 연속된 시퀀스 상에서 몇 프레임이상 중첩된 중심좌표를 중심으로 객체 추적 알고리즘을 구현하면 좀 더 정확한 객체 영역의 추적이 가능한데, 기존의 추정 방법을 이용할 경우는 제안된 추정 방법을 이용한 경우와는 달리 객체 영역의 중첩 중심 좌표가 객체 추적에 필요한 수보다 훨씬 적게 나타남에 따라 그림 11의 (c)에서와 같이 객체 영역 추적이 제대로 행해지지 못하는 결과를 나타내고 있다.

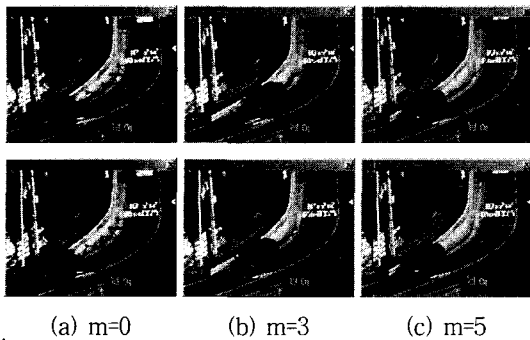


그림 11. 움직이는 객체의 중첩 중심 좌표,  $m$ 과 그에 따른 경로의 추적(위-제안된 벡터 영역 하에서, 아래-기존의 벡터 영역 하에서)

Fig. 11. Object tracking with overlapped object center position,  $m$ . (Top-in the vector space by the proposed method, Bottom-in the vector space by the conventional method).

연속된 프레임 상에서 객체 영역의 중심 좌표가 같은 위치에 중첩된 횟수를  $m$ 이라고 할 때, 그림 11의 (a)는  $m=0$ 인 상황, 즉, 중심 좌표의 중첩 횟수를 고려하지 않고 모든 프레임으로부터 얻어진 객체 영역의 중심 좌표로부터 객체의 움직임을 추적한 것이다. 기존

의 벡터 추정 기법을 이용한 벡터 영역 뿐 아니라 제안된 벡터 추정 영역에서도 객체 영역 내 움직임 벡터의 변화로 인한 객체 중심 좌표의 심한 요동으로 효과적인 객체 추적이 수행되지 못하는 모습을 보이고 있다. 그러나, 그림 11의 (b)와 (c)에서는, 제안된 벡터 영역에서의 객체 추적 알고리즘이  $m$ 이 일정 수준 증가할 수록 얻어진 객체의 중첩 중심 좌표로부터 효과적으로 움직임 객체를 추적하는 반면에, 기존의 알고리즘을 적용한 벡터 영역에서는 중첩된 중심 좌표를 제대로 추출해 내지 못함으로써 객체를 적절하게 추적하지 못하는 결과를 나타내고 있다.

제안된 방법으로부터의 이러한 성능은 표1에서 이미 나타난 바와 같이 기존의 벡터 재추정 방법에서보다 크게 늘어난 움직임 시퀀스에서의 객체 검출 프레임 수가 정확한 움직임 객체의 추적 알고리즘에 매우 효과적으로 기여하고 있음을 잘 보여준다고 할 수 있다.

## VI. 결 론

지금까지, MPEG 비디오에서의 움직임 벡터 재해석을 통한 효과적인 객체 움직임 검출 알고리즘을 제안하였다. 실험 결과에서 보여주는 것과 같이 제안된 움직임 벡터 재해석 방법은 기존의 방법과 비교하여 객체의 효과적인 추출 및 추적 기법 등 비디오에서의 다양한 움직임 해석을 위한 효과적인 성능을 좀 더 잘 나타내고 있다. 이러한 움직임 벡터 재해석 기법은 비단 객체의 추적 기법에서 뿐만 아니라 비디오 시퀀스에서의 전역적인 카메라 움직임 검출과 비디오 인덱싱이나 검색을 위한 특징 정보로서도 폭 넓은 활용이 가능할 것이다.

## 참 고 문 헌

- [1] Y. Nakajima, K. Ujihara and A. Yoneyama, "Universal scene change detection on MPEG-coded data domain," in Proc. SPIE Visual Comm. and Image Proc., vol. 3024, pp. 992-1003, 1997.
- [2] N.V. Patel and I.K. Sethi, "Video shot detection and characterization for video databases," Pattern Recognition, vol. 30, no. 4, pp. 583-592, 1997.

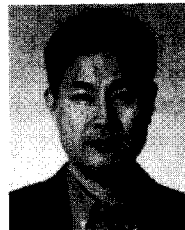
- [3] H.L. Eng and K.K. Ma, "Bidirectional motion tracking for video indexing," *Multimedia Signal Processing*, pp. 153-158, 1999.
- [4] J.G. Kim, H.S. Chang, J.W. Kim and H.M. Kim, "Efficient camera motion characterization for MPEG video indexing," *IEEE International Conference on Multimedia and Expo*, vol. 2, pp. 1171-1174, 2000.
- [5] O. Sukmarg and K.R. Rao, "Fast object detection and segmentation in MPEG compressed domain," *Proceedings of TENCON 2000*, vol. 3, pp. 364-368, 2000.
- [6] A. Yoneyama, Y. Nakajima, H. Yanagihara and M. Sugano, "Moving object detection and identification from MPEG coded data," *Proc. of International Conference on Image Processing*, vol. 2, pp. 934-938, 1999.
- [7] F. Bartolini, V. Cappellini and C. Giani, "Motion estimation and tracking for urban traffic monitoring," *Proc. of International Conference on Image Processing*, vol. 3, pp. 787790, 1996.
- [8] ISO/IEC/JTC1/SC29/WG11, 13812-2, 1995.
- [9] R. Milanese, F. Deguillaume and A. Jacot-Descombes, "Video segmentation and camera motion characterization using compressed data," *SPIE-Multimedia Storage and Archiving Systems II*, vol. 3229, pp. 23-31, 1997.
- [10] V. Kobla and D. Doermann, "Compressed domain video indexing techniques using DCT and motion vector information in MPEG video," in *Proc. SPIE Conference on Storage and Retrieval for Image and Video Database*, vol. 3022, pages 200-210, 1997.

저 자 소 개



金 洛 馬(學生會員)  
 1997년 2월 : 중앙대학교 제어계측 공학과 졸업(공학사). 2002년 2월 : 중앙대학교 첨단영상대학원 영상공학과 졸업(공학석사). 2002년 3월~현재 : 중앙대학교 첨단영상대학원 영상공학과 박사과정 재학중.

<주관심분야 : 영상처리, 영상정보기술>



金 泰 鎔(正會員)  
 1986년 2월 : 한양대학교 전기공학과 졸업(공학사). 1988년 2월 : 한양대학교 전자통신공학과 졸업(공학석사). 1998년 2월 : 포항공과대학교 컴퓨터공학과 졸업(공학박사). 2000년 1월~현재 : 중앙대학교 첨단영상대학원 교수.

<주관심분야 : 컴퓨터비전, 컴퓨터보안, 영상통신, 영상처리>



姜 應 寬(正會員)  
 1993년 2월 : 중앙대학교 전자공학과 졸업(공학사). 1995년 2월 : 중앙대학교 전자공학과 졸업(공학석사). 1995년 1월~1997년 5월 : 현대전자정보통신연구소 연구원. 2001년 2월 : 중앙대학교 첨단영상대학원 영상공학과 졸업(공학박사).

2001년 9월~현재 : 전주대학교 정보기술컴퓨터공학부 멀티미디어공학과 교수. <주관심분야 : 동영상부호화, 멀티미디어신호처리, 멀티미디어데이터베이스, 영상통신>



崔 宗 秀(正會員)  
 1975년 2월 : 인하대학교 전기공학과 졸업(공학사). 1977년 2월 : 서울대학교 전자공학과 졸업(공학석사). 1981년 2월 : 일본 Keio University 전기공학과 졸업(공학박사). 1981년 9월~1999년 8월 : 중앙대학교 전자공학과 교수.

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