

# MPEG 에서의 움직임 벡터 재구성을 이용한 비디오 해석 기법 연구

김낙우, 김태용, 강웅관, 최중수  
중앙대학교 첨단영상대학원

## A Study on Video Analysis using Re-constructing of Motion Vector on MPEG Compressed Domain

N. W. KIM, T. Y. KIM, E. K. KANG, J. S. CHOI

Department of Image Engineering, Graduate School of  
Advanced Imaging Science, Multimedia, and Film, Chung-Ang University  
mysope@imagelab.cau.ac.kr

### 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 재 추정된 각 프레임에서의 움직임 벡터를 이용한 비디오 시퀀스 내에서의 객체 추출 및 추적 기법 등에 대해서도 함께 제안하였다. 제안된 알고리즘은 영상에 대한 복원과정을 거치지 않고, 압축 비디오 영역으로부터 쉽게 추출될 수 있는 매크로 블록 영역 상에서 수행되었으며, 실험 결과는 제안된 방법의 높은 성능을 잘 나타내어 주고있다.

### 1. Introduction

MPEG 시퀀스의 움직임(motion)은 비디오 파싱을 통해 분할된 샷의 내용을 이해하고 샷에 내재된 장면 및 카메라에 의한 움직임 등을 분석하는 데에 있어서 매우 중요한 성분 요소이다. 최근 지능적인 비디오 브라우징/분석 등의 목적을 위해, 이러한 샷에서의 움직임 특징을 이용한 객체의 분할, 추적 방법 등이 활발히 연구되고 있다[1][2].

본 연구에서는, 샷에서의 이러한 움직임을 이해하는 데에 있어서, MPEG 압축 영역에서의 움직임 벡터 정보를 이용한 비디오에서의 효율적인 움직임 해석 방법에 대해 제안하고 있다. 우선, MPEG 시퀀스로부터 추출된 움직임 벡터를 효과적으로 재구성함으로써 프레임에서의 특징 정보로서의 사용을 가능하게 하였고, 영상 내 객체의 좌표 및 움직임 각도 등을 고려하던 기존의 알고리즘[3][4]과는 달리 움직이는 객체의 속도와 가속도 등을 고려한 새로운 객체 추출 및 추적 알고리즘을 또한 제시하고 그 성능을 비교하였다. 특히, 제안된 알고리즘은 MPEG 에서의 움직임 벡터 정보만을 이용함으로써 전체

영상의 디코딩(decoding)에 따른 시간적인 손실을 크게 줄일 수 있다.

본 논문의 순서는 다음과 같다. 우선, 2 장에서는 MPEG 에서의 움직임 해석을 위한 기존 알고리즘의 문제점을 살펴보고, 이의 해결을 위한 새로운 알고리즘을 제시한다. 3 장에서는 2 장에서 제안된 움직임 재추정 알고리즘을 이용한 압축영역에서의 객체 추출 및 추적 기법 등을 제안하며, 마지막으로, 4 장에서 이에 대한 실험 결과 및 결론을 도출한다.

### 2. Motion Flow Analysis

일반적으로, 압축 스트림에서의 매크로 블록은 프레임의 유형에 따라 각기 다른 수의 움직임 벡터를 가진다. 이는 그 매크로 블록이 인트라 부호화 블록인지, 순방향이나 역방향 혹은 양방향 예측 블록인지에 따라 달라지게 된다. 결국 이러한 압축영역에서의 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.

#### 2.1 일반적인 벡터 추정 방법

MPEG 시퀀스의 GOP 구조 상에서 두개의 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[5][6]. 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 알고리즘을 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.  $B_1$  프레임에서의 매크로 블록  $MB(u,v)$ 가 움직임 벡터  $(x,y)$ 를 갖는다는 의미는  $R_i$  프레임에서의 매크로 블록  $MB(u,v)$ 가  $B_1$  프레임으로 움직임 벡터  $(x,y)$  만큼 이동했다는 것을 나타내므로, 역으로 생각해보면  $R_i$  프레임에서의 매크로 블록  $MB(u,v)$ 는  $B_1$  프레임에서의 매크로 블록  $MB(u,v)$ 에 대응하는 움직임 벡터  $(-x,-y)$ 를 갖는다고 생각해보 수 있다. 이를 이용하여 참조 프레임  $R_i$  에서의 움직임 추정 벡터를 얻을 수 있다. 두 번째 단계는 참조 프레임  $R_j$  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  $R_j$  프레임에 대한  $B_n$  프레임에서의 벡터는 역방향 움직임 벡터이므로 벡터를 다시 역으로 변환할 필요는 없다[6].

그러나, GOP 구조에서의 이러한 벡터 추정 방식은 각각의 프레임을 나타내기에 충분할 만큼의 유효 움직임 벡터를 추출하지 못하다는 문제점을 가지고 있다. 위의 서술에서 이미 나타난 바와 같이,  $R_i$  프레임에서의 움직임 벡터 추정은  $B_1$  프레임 상에서의 순방향 움직임 벡터를 이용하고,  $B_n$  프레임에서의 움직임 벡터 추정은  $R_j$  프레임으로부터 역방향 예측된  $B_n$  프레임 상에서의 역방향 움직임 벡터만을 고려하고 있다. 이는 MPEG 시퀀스에서의 프레임 구조가 쌍방향 예측이 가능하다는 것을 간과한 것인데,  $B_n$  프레임이 비단  $R_j$  프레임에서의 역방향 움직임 벡터 뿐만 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.

## 2.2 제안된 벡터 추정 방법

이러한 문제점을 보완하기 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 쌍방향 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  $R_i$  프레

임에서의 예측 벡터의 수를 크게 늘릴 수 있다.  $B_n$  프레임에서도 이와 같은 방법을 이용한다. 단,  $R_j$  프레임이 I 픽처일 경우  $R_i$ 에서  $R_j$ 로의 순방향 벡터를 찾을 수 없으므로 이러한 방법을 이용할 수 없다.

이 알고리즘의 수식적 표현은 수식 1 과 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 매크로 블록의 순방향 움직임 벡터를  $\overline{R_j R_i}$  로 할 때,  $R_j$  프레임에서  $B_n$  프레임으로의 역방향 추정 벡터  $\overline{B_n R_j}$  은 다음의 수식으로 쉽게 얻을 수 있다.

$$\overline{B_n R_j} = \overline{B_n R_i} - \overline{R_j R_i} \quad (1)$$

그리고,  $B_1$  프레임 상에서 현재 매크로블록의 역방향 움직임 벡터를  $\overline{B_1 R_j}$  로 했을 때,  $B_1$  프레임에서  $R_i$  프레임으로의 역방향 추정 벡터  $\overline{R_i B_1}$  의 계산 역시 쉽게 얻어진다.

$$\overline{R_i B_1} = -\overline{R_j R_i} - \overline{B_1 R_j} \quad (2)$$

이와 같은 방법으로 기존의 알고리즘을 보완할 경우  $B_1$  이나  $B_n$  프레임에서의 예측 벡터의 수를 기존의 알고리즘에 비해 크게 늘릴 수 있고, 이를 통해 프레임에서의 효과적인 특징 정보로서 이러한 예측 벡터들을 사용할 수 있게 된다.

## 3. Moving Object Tracking on Compressed Domain

본 장에서는 2 장에서 제안된 방법을 통해 추출되어진 각 프레임에서의 움직임 추정 벡터를 이용하여 객체의 움직임을 가지는 연속적인 프레임에서의 객체의 분할 및 추적에 대한 알고리즘을 제안한다.

### 3.1 Moving object extraction

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 2 차원 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 비교함으로써 손쉽게 제거가 가능하다. 그리고 나서, 데이터 취득과정의 오류나 환경적 요인 등으로 인해 발생된 노이즈를 제거하기 위해 배경이 제거된 벡터 공간상에 백

터 미디언 필터링 을 적용시킴으로써 전체 벡터 영역을 스무딩(smoothing) 한다. 이러한 과정을 거쳐 얻어진 영역이 일정한 크기 이상일 때, 최종적인 객체 영역으로 분리되어진다.

### 3.2 Moving Object tracking

움직이는 객체의 추적은 영상 시퀀스에서의 시공간적 관계에 의해서 수행된다. 임의의 시퀀스에서 동일한 객체는 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 이전프레임에서의 영역과 비교, 링크시킴으로써 시퀀스에서의 전체적인 움직임 궤적을 추적할 수 있다.

움직임 객체의 추적 알고리즘은 크게 두 부분으로 나뉘어 진다. 첫 단계는, 연속된 두 프레임 A, B에 각각 속해있는 영역들의 공간상의 중심좌표를 이용, 이들 영역 간의 중심점간 최소거리 계산을 통해 프레임간 대응 영역을 찾는다. 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 생각되어지는 이전 영역으로부터 영역의 최대 속도와 가속도를 유추하여 대응점  $C_p$  를 구하고 이 대응점과 이전, 현재 영역에서의 영역 중심점  $C_{i-1}, C_i$  와의 길이를 서로 비교함으로써 영역의 연속성을 검증하였다. 그림 1 은 이전 영역으로부터의 선형적인 객체 움직임을 고려한  $C_p$  점 탐색 방법을 도시하고 있다. 영상에서의 객체의 움직임이 일정한 선형성을 갖는 것을 고려하였을 때, 식 (4)에서 좌표  $C_p$  는 좌표  $C_{i-1}$  과 서로 비례하므로  $C_i$  의 좌표가 급격히 변하지 않는 한, 식 (3)의 수식은 항상 만족되어지게 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 (3)을 만족하게 된다. 우리는 이와 같은 두 단계를 거쳐 이전 영역과 현재 영역과의 객체의 연속성을 파악하였다. 두 영역에서의 중심 각도의 흐름과 움직임 벡터의 수의 유사성까지 고려되어진다면 좀 더 정확한 객체 추적이 가능해 질 수 있다.

$$\left| \overline{C_p C_i} - \overline{C_p C_{i-1}} \right| < \beta \cdot (v+a) + \gamma \quad (3)$$

$$C_p = C_{i-1} + v + a \quad (4)$$

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

$$t = i-k, i-k+1, \dots, i-1$$

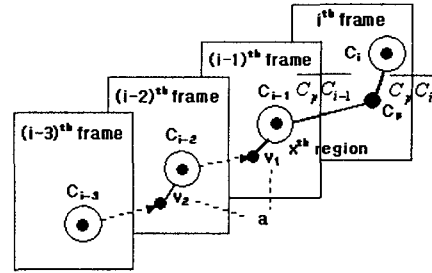


그림 1. 이전 영역의 움직임을 고려한 객체 추적

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.  $v$  와  $a$  는 각각 이전 영역의 평균 속도와 가속도를 나타내며 정수의 값을 갖는다.  $\beta$  는 비례 정수이고 객체의 움직임이 선형적이라고 가정할 때 통상적으로 2 의 값을 갖게 된다.  $\gamma$  는 옵션값으로  $v+a$  의 값이 0 일 때는 2 의 값을 갖고, 그 외 일 때는 0 의 값을 갖는다.

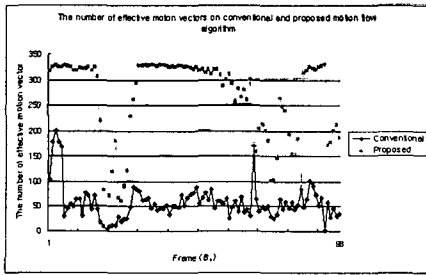
본 장에서의 이러한 알고리즘의 성능은 본질적으로 각각의 프레임을 묘사하는 움직임 벡터의 정확성과 유효 벡터의 수 등에 매우 민감한 영향을 받게 된다. 결국 2 장에서 제안된 각 프레임에서의 효과적인 벡터 추정 방법이 본 장에서 제안된 알고리즘의 성능 향상에 큰 영향을 미친다.

### 4. Experimental Results and Conclusions

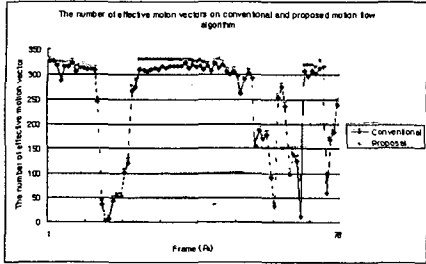
이 장에서는 2 장에서 제안된 움직임 벡터 추정 기법에 대한 유효성 검토와 3 장에서 제안된 객체의 움직임 추출 알고리즘에 대한 성능을 제시한다.

그림 2 는 기존의 움직임 추정 방법과 제안된 방법에서의 유효벡터 수를 비교하여 보여주고 있다.  $B_n$  프레임에서의 추정 움직임 벡터는 제안된 방법에서 3 배 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.

$R_i$  프레임에서는  $B_n$  프레임 보다 상대적으로 적은 10% 정도의 유효벡터 성분의 증가가 나타나는데 이는 일반적인  $B_1$  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.



(a)  $B_n$  프레임에서의 움직임 추정 벡터의 수



(b)  $R_i$  프레임에서의 움직임 추정 벡터의 수

그림 2. 기존 방법과 제안된 방법에서의 유효 움직임 추정 벡터 수 비교

또한, 움직임 객체의 추출을 위해 3.2 절의 알고리즘을 제안된 알고리즘을 이용한 벡터 추정 영역에 적용할 시, 일반적인 알고리즘을 이용한 벡터 추정 영역에 적용할 때와 비교하여 객체가 검출된 프레임의 비율이 30% 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 것이다.



(a) 일반적인 벡터 추정 방법을 이용한 객체 추적



(c) 제안된 벡터 추정 방법을 이용한 객체 추적

그림 3. 벡터 추정 방법에 따른 움직임 객체 추적 결과 비교

MPEG 시퀀스에서의 이러한 객체 검출 프레임 수의 증가는 그림 3에 나타난 바와 같이 움직임 객체의 추적에도 그 영향을 미친다. 3.2 절에서 제안된 객체 추적 알고리즘을 동일 적용하였을 때, 제안된 벡터 추정 영역 하에서의 움직임 객체의 추적은 그림 3-(a)에서 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 30% 이상 줄어든 객체 검출 프레임 수로 인해 생기는 객체 중심 영역의 불연속성에 기인한다. 즉, 중간 프레임의 공백으로 인해 생기는 객체 중심 좌표의 급격한 움직임이 비선형적 특성을 나타내기 때문이다.

지금까지, MPEG 비디오에서의 움직임 벡터 재해석을 통한 효과적인 객체 움직임 검출 알고리즘을 제안하였다. 실험 결과에서 보여주는 것과 같이 제안된 움직임 벡터 재해석 방법은 비디오에서의 다양한 움직임 해석을 위한 효과적인 성능을 잘 나타내고 있다. 이러한 알고리즘은 또한 비디오의 인덱싱이나 검색을 위한 특징 정보로서도 그 활용이 가능 할 것이다.

## 5. Acknowledgement

본 연구는 교육부의 두뇌한국 21 사업(BK21) 및 과기부의 국가지정 연구실(2000-N-NL-01-C-285), 그리고 정보통신부의 정보통신 우수시범학교 지원사업에 의해 수행되었습니다.

## References

- [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," *IEEE International Conference on Multimedia and Expo*, vol. 2, pp. 1171-1174, 2000.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.