

## 압축 도메인에서의 영상내 이동 물체 검출 기법

이봉렬, 이명진

한국항공대학교

counter83@gmail.com, artistic@kau.ac.kr

### A Study on a Detecting Method of Moving Objects in Compressed Domain

Bong-Ryul Lee, Myeong-jin Lee

Korea Aerospace University

#### 요약

최근 감시시스템 구축에서 디지털 비디오 레코더(DVR)를 이용하는 방식이 주를 이루고 있다. 영상압축 기술의 발전에 따라 디지털 비디오 레코더는 H.264/AVC 인코더를 기반으로 저장 공간 대비 고품질의 영상 저장이 가능해졌다. 또한, 다수 채널의 동시 처리가 가능해졌을 뿐만 아니라 영상 압축 및 저장 기능 외에 지능형 감시를 위한 기능이 요구되고 있다.

본 논문에서는 H.264/AVC 압축 비트열에서 움직임 벡터 추출 및 분석을 통한 영상내 이동 물체 검출 방법을 제안한다. 압축 비트열에서 현재 프레임과 이전 프레임의 움직임 벡터들의 크기와 각도를 비교하여 임계값 이하인 움직임 벡터들을 추출하여 이동 물체를 검출하였다. 이동 물체 검출 성능 평가를 위해 H.264/AVC의 참조 프로그램인 JM 15.1을 사용하여 인코딩 및 비트열 분석을 수행 하였고, 움직임이 검출된 시간 구간만을 디코딩 후 저장하는 알고리즘을 구현하였다. 제안된 방식은 디코딩 부분의 연산량 감소뿐만 아니라 저장소의 물리적 공간 절약이 가능하다.

#### 1. 서론

영상에서 물체를 검출하고 추적하기 위하여 다양한 연구들이 수행되어 왔다. 기존의 많은 연구들은 픽셀 도메인에서 영상 처리를 수행하였다. 그러나 최근의 일반적인 동영상들은 보통 부호화 과정을 통해 압축 도메인으로 변환 되어진 후 전송되기 때문에, 기존의 방법을 적용하기 위해서는 수신되어진 부호화 영상을 다시 복호하여 픽셀 도메인으로 변환 후 처리하는 과정이 필수적이다.

그러나, 최근 동영상 압축 코덱으로 많이 사용되어지는 H.264/AVC 코덱은 부호화 과정에서 움직임 벡터, DCT 계수 등의 정보가 생성되며, 이는 이동 물체의 검출 및 추적에 사용될 수 있다. 따라서, 이 정보들을 이용하면 부호화된 영상의 복호 없이 전송 정보의 분석을 통해 이동 물체의 검출이 가능하다[1-2]. 이와 같은 압축 도메인 기법은 픽셀 도메인 기법에 비해 연산량을 감소시킬 수 있다. 카메라에서 연산을 처리시 카메라 단말의 가격을 감소시킬 수 있으며, 다수 카메라를 사용하고 중앙 서버에서 연산하는 경우에는 연산의 부하를 대폭 줄일 수 있다. 또한, 이동 물체가 감지된 구간만을 저장한다면 데이터 저장소 자원의 절약이 가능할 수 있다.

기존 진행된 압축 도메인에서의 연구는 이동 물체의 검출에 초점을 두고 진행되었다[3-5]. 먼저 압축된 비트열을 분석하여 움직임 벡터와 DCT 계수 등을 추출하고 이를 가공하여 이동 물체를 검출하였다. 현재 프레임의 정보만을 고려하는 기존의 방식은 이동 물체가 정지하였을 경우 연속적인 검출 및 추적이 불가능 하다. 지속적인 물체의 검출은 감시시스템에서 요구되는 주요 사항이기 때문에, 해당 문제를 해결하기 위한 연구가 필요하다.

본 논문에서는 영상내 물체를 검출하고 추적하기 위한 응용을 위하여 압축 도메인 상의 이동 물체 검출 알고리즘을 제안한다. H.264/AVC 비트열 상의 움직임 벡터의 시간에 따른 변화 정도를 분석하여 이동 물체의 존재 여부를 결정한다. 본 연구에서는 일정 시간 윈도우 상에서 이동 물체가 정지한 경우에도 지속적으로 물체를 검출하기 위해 움직임 벡터 누적 기법을 제안하고, 실험을 통해 성능을 평가한다.

#### 2. 움직임 벡터 분석을 통한 이동 물체 검출

본 절에서는 비트열 분석을 통해 획득한 움직임 벡터를 사용하여 이동 물체를 검출하는 방법에 대해 설명한다. 먼저 참조 블록의 위치가 블록들의 경계에 나타나는 문제를 해결하고자, 4x4 블록을 대표하는 하나의 움직임벡터를 설정한다. 이후, 참조 블록과 현재 블록의 움직임 벡터 대표 값의 크기와 각도를 비교하여 이동 물체 검출에 사용한다.

##### 가) 압축 비트열상의 움직임 벡터 검증

부호화의 움직임 예측 단계에서는 매크로블록 단위로 참조화면 상의 최소 에러 블록을 현 블록의 참조 블록으로 결정한다. 이때 사용되는 매크로블록들의 참조 블록의 위치는 매크로블록 단위와 정확히 일치하지 않고 다른 매크로블록 사이의 경계에서 결정될 수도 있다. 본 논문에서는 4x4 블록단위 연산을 수행하므로 참조 블록과 현재 블록의 움직임 벡터의 직접 비교가 불가능하다. 이를 해결하기 위하여 블록 내 움직임 벡터의 대표 값을 결정하는 과정이 반드시 필요하다. 블록  $B_2$ 의 참조 블록  $B_1$ 이 그림 2와 같이 위치할 경우  $B_1$ 의 움직임 벡터

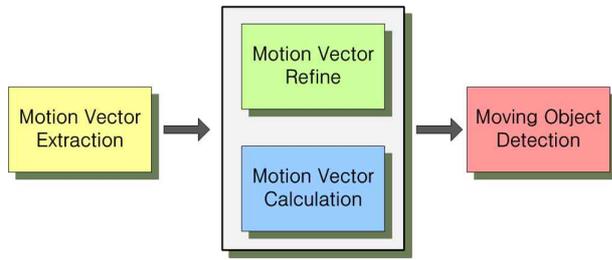


그림 1. 제안된 방법의 흐름도

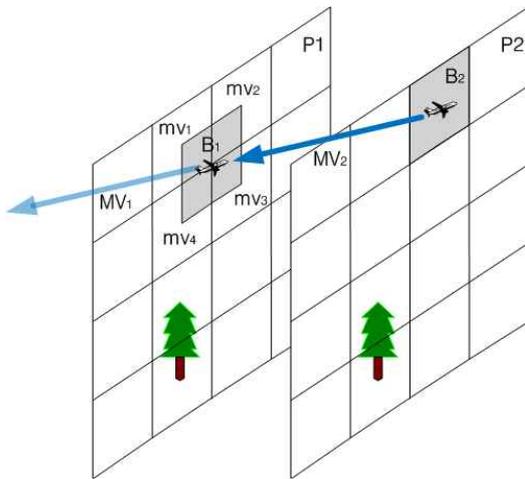


그림 2. 블록 내 움직임 벡터 대표 값 결정 과정

는 식 (1)과 같이 결정된다.

$$MV_B^t = \frac{1}{S_n^t} \sum_{m \in N} (MV_m^t \cdot S_m^t) \quad (1)$$

여기에서  $N$ 은 블록  $B$ 가 걸쳐진 블록의 집합이고,  $MV_m^t, S_m^t$ 는 각각 블록  $B$ 가 걸쳐진 주변 블록의 움직임 벡터와 블록 면적을 나타낸다.  $S_n^t$ 는 주변  $S_m^t$ 의 면적의 합이다.

H.264/AVC는 부호화 시 다양한 종류의 블록 크기를 사용한다. 본 연구에서는 프레임 내에서 블록 단위 처리를 용이하게 하기 위하여 움직임 벡터 처리 시 블록의 크기를 4x4로 통일하여 사용하였다. GOP(group of picture) 구조를 I, P1, P2, P3, ... I 라고 하면, P1 프레임의 움직임 벡터는 전부 참인 것으로 가정한다. 이는 현재 P1 프레임의 움직임 벡터와 비교할 I 프레임에 움직임 벡터가 존재하지 않기 때문이다. 이로 인해 실질적인 움직임 벡터의 검증은 부호화에 3번째로 들어오는 P2 프레임에서부터 적용이 된다. 움직임 벡터의 각도와 길이를 각각  $Ang(MV_n^t), Len(MV_n^t)$ 라고 하면, 움직임 벡터 검증을 위한 조건은 식 (2)과 같다.

$$\begin{cases} |Ang(MV_B^{t-1}) - Ang(MV_B^t)| < 90^\circ \\ |Len(MV_B^{t-1}) - Len(MV_B^t)| < 8(pixels) \end{cases} \quad (2)$$

식 (2)의 두 조건을 모두 만족하는 움직임 벡터들은 인접 프레임에서 비교적 일정한 방향과 크기를 가지게 되므로 실제 움직임으로 판단한다. 설정된 임계값을 벗어나는 움직임 벡터들은 이후의 단계에서 고려하지 않았다.

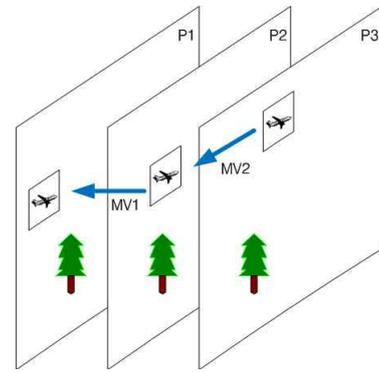


그림 3. 이동 물체 검증을 위한 움직임 벡터 구성

### 나) 이동 물체 검출

움직임 벡터를 검증하여 이동 물체를 감지하면, 현재 프레임 내에서 이동 물체의 공간 좌표 획득이 가능하다. 공간상의 같은 좌표에 연속해서 움직임 벡터가 검출된다는 것은 이동 물체가 존재함을 의미한다. 본 연구에서는 움직임 벡터 검증을 통해 획득한 좌표들의 위치를 누적하고 그 값의 변화에 따라 이동 물체를 검출한다. 움직임 검출 빈도 값  $val$ 는 식 (3)과 같이 결정된다.

$$val = 30x - 10(t - x) \quad (3)$$

식 (3)에서  $x$ 는 해당 위치에서 움직임 벡터가 검출된 횟수이며, 이동 물체가 검출 되는 빈도에 따라  $val$  값을 결정하였다. 움직임 검출 빈도값  $val$ 에 따른 누적 움직임 벡터를 식 (4)와 같이 정의한다.

$$acc\_mv = \begin{cases} 0, & \text{if } (val > th) \\ val, & \text{otherwise} \end{cases} \quad (4)$$

$acc\_mv$ 는 화면 표시를 위한 밝기 값으로 임계값  $th$ 에 의해 결정되어 0에서 255사이의 값을 갖는다.

### 3. 모의실험

알고리즘 성능 평가를 위해 실제 감시 시스템과 유사한 상황의 영상을 통해 실험을 진행하였다. 실험에 사용된 영상들은 H.264/AVC의 참조 프로그램인 JM 15.1을 이용하여 부호화된 후 사용되었다. 부호화 시 B 프레임은 사용하지 않고 I와 P 프레임만을 사용하였으며, 영상은 QVGA, 30 fps으로 촬영되었다.

그림 4의 영상 내 상황은 이동 중이던 사람 중 한 명이 방향 전환을 위해 잠시 정지한 상황이다. 그림 4의 (a), (b), (c)는 각각 원본영상, 움직임 벡터 영상, 누적 움직임 벡터 영상을 나타낸다. (c)를 통해 이동 중인 사람뿐만 아니라 일정시간 정지한 사람도 지속적으로 검출하는 것이 가능함을 확인 할 수 있다. 일시 정지한 물체의 검출은 식(3)의 임계값  $th$ 의 조절을 통해 가능하다.  $th$ 를 높게 설정하면 일시 정지한 물체의 검출 가능 시간이 증가하고,  $th$ 를 낮게 설정하면 일시 정지한 물체의 검출 가능 시간이 감소하게 된다. 적절한 파라미터 값의 설정을 통해 일시 정지 후 다시 이동하는 물체의 지속적인 감시가 가능함을 확인 할 수 있다.

### 4. 결론

본 연구에서는 H.264/AVC 압축 도메인 분석을 통하여 이동 물체

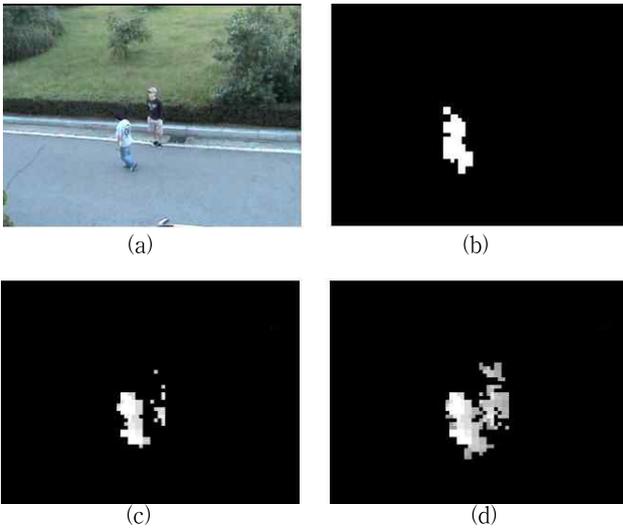


그림 4 (a) 원본영상, (b) 움직임 벡터 영상, (c),(d) 누적 움직임 벡터 영상, (c)는 임계값 200, (d)는 임계값 150 사용

를 검출하는 방법을 제안하였다. 이 방법은 압축된 형태로 네트워크를 통해 전송되는 감시 카메라 시스템에 적합하며, 기존의 픽셀 도메인의 이동 물체 검출 방식에 비하여 연산량과 데이터 저장 공간의 감소를 가져 올 수 있다. 또한, 기존 연구에 비하여 이동 물체가 일정시간 정지한 경우에도 지속적으로 관찰이 가능한 특징을 가지고, 실험을 통해 검증하였다. 추후 연구에서는 교차 및 겹침 현상 극복을 위한 방법과 다중 카메라 연계를 통한 이동 물체 검출 추적 연구가 필요하다.

## Acknowledgment

본 연구는 경기도지역협력연구센터(GRRC) 프로그램에 따른 한국항공대학교 차세대방송미디어기술 연구센터(GRRC항공2010-B03)와 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2010-0012390) 지원을 받아 수행된 것임.

## 참 고 문 헌

- [1] F. Porikli, F. Bashir, S. Huifang, "Compressed Domain Video Object Segmentation", *IEEE Trans. on CSVT*, vol. 20, Issue 1, pp 2-14, 2009.
- [2] W. Wang, J. Yang, W. Gao, "Modeling Background and Segmenting Moving Objects from Compressed Video", *IEEE Trans. on CSVT*, vol. 18, Issue 5, pp 670-681, 2008.
- [3] S. K. Kapotas, A. N. Skodras, "Moving object detection in the H.264 compressed domain", *IST*, pp. 325-328, 2010.
- [4] S. Verstockt, S. D. Bruyne, C. Poppe, P. Lambert, R. Walle, "Multi-view Object Localization in H.264/AVC compressed Domain", *AVSS*, pp. 370-374, 2009.
- [5] Z. Qiya, L. Zhicheng, "Moving Object Detection Algorithm for H.264/AVC Compressed Video Stream", *CCCM*, pp 186-189, 2009.