

정규논문 (Regular Paper)
방송공학회논문지 제20권 제3호, 2015년 5월 (JBE Vol. 20, No. 3, May 2015)
<http://dx.doi.org/10.5909/JBE.2015.20.3.449>
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

HEVC 스트림 상에서의 객체 추적 방법

박 동 민^{a)}, 이 동 규^{a)}, 오 승 준^{a)†}

Object Tracking in HEVC Bitstreams

Dongmin Park^{a)}, Dongkyu Lee^{a)}, and Seung-Jun Oh^{a)†}

요 약

동영상에서의 객체 추적은 보안, 색인 및 검색, 감시, 통신, 압축 등 다양한 분야에서 중요하다. 본 논문은 HEVC 비트스트림 상에서의 객체 추적 방법을 제안한다. 복호화를 수행하지 않고, 비트스트림 상에 존재하는 움직임 벡터(MV : Motion Vector)와 부호화 크기 정보를 Spatio-Temporal Markov Random Fields (ST-MRF) 모델에 적용해 객체 움직임의 공간적 및 시간적 특성을 반영한다. 변환계수를 특징점으로 활용하는 객체형태 조정 알고리즘을 적용해 ST-MRF 모델 기반 객체 추적방법에서 나타나는 과분할에 의한 오차전파 문제를 해결한다. 제안하는 방법의 추적성능은 정확도 86.4%, 재현율 79.8%, F-measure 81.1%로 기존방법 대비 평균 F-measure는 약 0.2% 향상하지만 기존방법에서 과분할 및 오차전파가 두드러지는 영상에 대해서는 최대 9% 정도의 성능향상을 보인다. 전체 수행시간은 프레임 당 평균 5.4ms이며 실시간 추적이 가능하다.

Abstract

Video object tracking is important for variety of applications, such as security, video indexing and retrieval, video surveillance, communication, and compression. This paper proposes an object tracking method in HEVC bitstreams. Without pixel reconstruction, motion vector (MV) and size of prediction unit in the bitstream are employed in an Spatio-Temporal Markov Random Fields (ST-MRF) model which represents the spatial and temporal aspects of the object's motion. Coefficient-based object shape adjustment is proposed to solve the over-segmentation and the error propagation problems caused in other methods. In the experimental results, the proposed method provides on average precision of 86.4%, recall of 79.8% and F-measure of 81.1%. The proposed method achieves an F-measure improvement of up to 9% for over-segmented results in the other method even though it provides only average F-measure improvement of 0.2% with respect to the other method. The total processing time is 5.4ms per frame, allowing the algorithm to be applied in real-time applications.

Keyword : Object tracking, Markov random fields, HEVC, compressed domain

a) 광운대학교 전자공학과(Dept. of Electronic Engineering, Kwangwoon university)

† Corresponding Author : 오승준(Seung-Jun Oh)

E-mail: sjoh@kw.ac.kr

Tel: +82-2-940-5102

ORCID: <http://orcid.org/0000-0002-5036-3761>

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원(No.R0101-15-293, 방송용 영상 인식 기반 객체 중심 지식융합 미디어 서비스 플랫폼 개발)과 2013년도 광운대학교 교내학술연구비 지원에 의해 연구되었음.

· Manuscript received March 23, 2015; revised May 12, 2015; accepted May 12, 2015.

1. 서론

동영상에서의 객체 추적(Video Object Tracking : VOT)은 보안, 색인 및 검색, 감시, 통신, 압축 등 다양한 분야에 활용된다. VOT에서 객체를 분할하고 추적하는 방식은 화소영역 기반 방식과 압축영역 기반 방식으로 나누어진다. 화소영역 기반 방식은 일반적으로 높은 정확도를 가지지만 높은 복잡도를 요구한다. 또한 오늘날 대부분의 비디오 콘텐츠는 압축된 비트스트림 (bitstream) 형태 즉, H.264/Advanced Video Coding(H.264/AVC)^[1] 또는 최근 표준화가 완료된 High Efficiency Video Coding(HEVC)^[2] 비트스트림으로 존재하기 때문에 화소 정보를 추출하는 복호화 과정이 필수적이다. 반면에 압축영역 기반 방식은 압축된 비트스트림 상에 존재하는 움직임 벡터(Motion Vector : MV)나 부호화 모드, 잔여신호, 변환계수 등의 정보를 사용한다. 화소 정보를 사용하지 않아 화소영역 기반 방식에 비해 정확도가 떨어지지만 복호화 과정을 생략할 수 있어 실시간 응용프로그램에 더 적합한 방식이다. 색인 및 검색 분야에서 검색 범위를 조기 결정하여 검색 시간을 단축할 수 있으며, 객체 검출 및 추적 분야에서 압축영역 기반 방식을 통한 대략적인 객체 검출 후 화소 정보를 활용한 정밀한 객체 검출 등의 활용이 가능하다. 영상 복호화 분야에서 영상의 관심영역을 복호화 이전에 미리 결정하여 선택적 복호화가 가능하다.

압축영역 기반 방식의 기존 연구로는 전역 움직임 추정(Global Motion Estimation : GME)과 매크로블록 제외(macroblock rejection) 방법을 이용하여 H.264/AVC 비트스트림 상에서 매크로블록 단위로 객체 추적을 수행하는 방법이 있다^[3]. 하지만 이 방식은 배경 움직임에 비해 객체의 움직임이 충분히 크지 않으면 추적하지 못하는 한계가 있다. Kas와 Nicolas는 객체의 궤적을 H.264/AVC Scalable Video Coding (SVC)의 움직임 벡터를 이용하여 추정하였다^[4]. 우선 배경분리 기술을 통해 전경에 존재하는 물체를 분리하고, 시간적 필터링(temporal filtering)을 통해 잡음을 제거한다. Timed motion history images 방법을 통해 객체 추적이 수행된다. H.264/AVC 비트스트림의 움직임 벡터와 분할 크기(partition size) 정보를 mean shift clustering을

이용하여 객체를 분리하는 접근법도 있다^[5]. 두드러진 움직임 벡터를 공간-시간적 중앙값 필터(median filter)와 전역 움직임 보상(Global Motion Compensation : GMC)을 적용하여 추출한 후, 공간적 mean shift를 통해 움직임이 균일한 지역을 찾아내고, 시간적 mean shift를 통해 그 지역을 부드럽게 해준다. 압축영역 기반 방식에서 마르코프 임의의 장(Markov Random Fields : MRF) 모델을 이용하여 객체 추적을 수행하는 연구들이 있다. Treetasanatavorn은 움직임 벡터를 깁스-마르코프 임의의 장 이론(Gibbs-MRF theory)과 베이저안 추정(Bayesian estimation)을 이용하여 객체 추적을 수행하였다^[6]. 첫 번째 프레임에 대한 객체 분리 작업은 확률적 움직임 일관성 모델(stochastic motion coherence model)^[7]을 통해 이뤄지고, 연속적인 프레임들에 대해서는 어파인 모델(affine model)을 이용하여 이전 프레임에 존재하는 분할 정보를 현재 프레임에 사영(projection)시켜 현재 프레임에 대한 분할 정보를 예측한다. 그 후, 최종적인 분할 라벨(label)을 부여한다. Zeng은 마르코프 임의의 장 분류(classification)를 적용하여 객체 추적을 수행하였다^[8]. 움직임 벡터들은 배경 에지, 전경, 잡음 중에서 한 형태로 분류된다. 그 후, 최대사후확률(maximum a posteriori probability : MAP)을 최대화 하는 방법으로 객체를 추출한다. Chen은 coarse-to-fine 분할 방법을 적용하여 객체 추적을 수행하였다^[9]. 비트스트림 상에서 MRF모델을 기반으로 객체 영역을 분할한 후 복호화한 화소정보를 이용해 에지(edge)를 찾아 region growing 시켜 분할된 객체의 형태를 개선한다. Mak은 RANSAC(RANdom SAMple consensus)과 MRF모델을 이용하여 객체 추적을 수행하였다^[10]. RANSAC을 이용해 전역 움직임 보상과정을 수행하고 움직임, 공간적 유사성, 공간적 분할연속성, 시간적 분할연속성의 총 4가지 특성을 기반으로 객체를 추출한다. 최근에는 Khatoonabadi가 공간-시간적 마르코프 임의의 장(Spatial-Temporal Markov Random Fields : ST-MRF) 모델을 사용하는 객체 추적 방법을 제안하였다^[11]. ST-MRF모델은 기존 MRF 모델에 객체의 시간적 양상을 추가적으로 반영하여 이를 기반으로 객체를 추출하지만 추가적인 고속화 여지를 지니고 있으며 객체의 형태에 대한 정보를 반영하지 못해 이로 인한 과분할(over-segmentation) 문제를 지니고

있다. 과분할 문제가 발생하면 연속된 영상에서 오분류된 객체정보를 참조하므로 추가적으로 오차전파(error propagation) 현상이 발생해 전체적인 추적성능이 저하된다.

본 논문에서는 최근 표준화가 완료된 HEVC 표준을 따르는 비트스트림 상에서의 객체 추적 방법을 제안한다. 기본적인 추적 모델은 Khatoonabadi의 ST-MRF 모델^[11]을 기반으로 하면서 추가적으로 HEVC 비트스트림 상에 존재하는 변환계수 정보를 이용해 객체의 형태를 반영하도록 하여 기존에 나타나는 과분할 문제를 해결한다. HEVC 비트스트림 상에 존재하는 움직임 벡터, 예측 단위의 크기, 변환 계수의 3가지 정보를 추적을 위한 특징점으로 활용한다. 움직임 벡터가 없는 블록에 움직임 벡터를 추정하는 전처리 과정을 거쳐 움직임 벡터를 할당한다. 전처리과정을 거치고 나면 전역 움직임 보상과정을 통해 카메라의 움직임 등으로 발생하는 전역 움직임(Global Motion : GM)을 제거하여 객체를 구분하며 이를 영상 전체가 아닌 관심영역 단위로 수행함으로써 알고리즘의 고속화를 얻는다^[12]. 구분된 객체에 대해 ST-MRF모델을 기반으로 객체를 추적한다. 후처리과정으로 변환계수 정보를 활용한 객체 형태 조정 알고리즘을 통해 객체의 형태를 조정한다.

본 논문의 구성은 다음과 같다. II장에서는 ST-MRF 모델 기반 객체 추적 방법을 간략히 기술한다. III장에서는 본 논문에서 제안하는 객체 추적 방법을 자세하게 기술한다. IV장에서는 제안하는 객체 추적 방법에 대한 실험 결과를 보이며 V장에서는 본 논문에 대한 결론을 맺는다.

II. ST-MRF 모델 기반 객체 추적 방법

본 논문에서 제안하는 객체 추적 방법은 Khatoonabadi가 제안한 ST-MRF 모델 기반 객체 추적 방법을 기반으로 한다^[11]. ST-MRF모델은 기존 MRF 모델에 객체의 움직임의 시간적 양상을 추가적으로 반영하여 움직임 일관성(motion coherence), 공간적 밀집도(spatial compactness), 시간적 연속성(temporal continuity)을 기반으로 H.264/AVC 비트스트림 상에서 객체를 추출한다.

첫 번째 프레임에서 사용자가 추적하고자 하는 객체를

직접 선택하면 연속된 프레임들에 대해서 자동으로 객체 추적이 이루어진다. 추적하고자하는 프레임에 대해서 비트스트림으로부터 움직임 벡터, 부호화 정보 2가지 정보를 추출해 특징점으로 활용한다.

추적을 수행하기 전에 전처리과정으로 움직임 벡터를 포함하지 않는 매크로블록의 움직임 벡터를 주변 움직임 벡터들을 이용하여 추정하고 전역 움직임 추정 및 보상과정을 통해 객체와 배경의 움직임을 분리한다. 그 후, ST-MRF 모델의 최대사후확률 해법을 Iterated Conditional Modes (ICM)^[13] 알고리즘을 통해 얻어 객체의 형태와 위치가 예측된다. Khatoonabadi가 제안한 방법의 전체 흐름도는 그림 1과 같다.

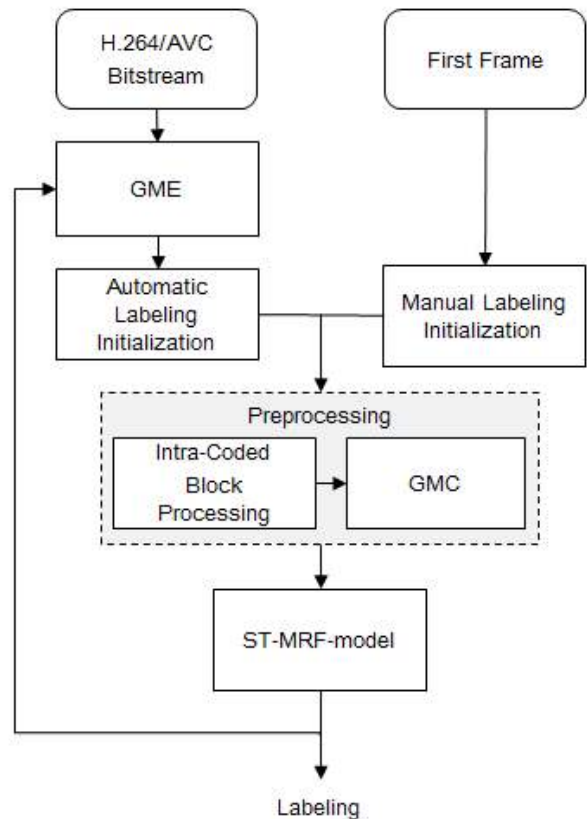


그림 1. ST-MRF 모델 기반 객체 추적 방법의 전체 흐름도
 Fig. 1. Flow chart of the ST-MRF model based object tracking method

그러나, 이 방식은 객체의 형태에 대한 정보를 반영하지

못해 배경과 객체가 동일한 움직임 벡터를 갖는 환경에서 과분할(over-segmentation) 및 오차전파(error propagation) 문제를 지니고 있다. 배경과 객체가 비슷한 움직임을 가질 경우 배경이 객체로 오분류 될 수 있다. 오분류로 인한 과분할 문제가 발생하면 연속된 영상에서 오분류된 객체정보를 참조하므로 오차전파 현상이 발생해 전체적인 추적 성능이 저하된다. 본 논문에서는 변환계수를 특징점으로 활용하여 객체 형태에 대한 정보를 반영함으로써 과분할 및 오차전파 문제를 효과적으로 해결한다.

III. 제안하는 객체 추적 방법

본 논문에서 제안하는 객체 추적 방법은 HEVC 표준 비트스트림을 대상으로 한다. 첫 번째 프레임에서 사용자가 추적하고자 하는 객체를 직접 선택하면 연속된 프레임들에 대해서 자동으로 객체 추적이 이루어진다. 추적하고자 하는 프레임에 대해서 HEVC 비트스트림으로부터 움직임 벡터, 예측 단위의 크기, 변환계수의 3가지 정보를

추출해 특징점으로 활용한다. 움직임 벡터는 객체의 움직임을 반영하고, 예측 단위의 크기와 변환계수는 객체의 경계를 나타낸다.

추적을 수행하기 전에 화면 내 예측 모드로 부호화되어 움직임 벡터를 포함하지 않는 부호화 단위(Coding Unit : CU)의 움직임 벡터를 주변 예측 단위들의 움직임 벡터들을 이용하여 추정하는 전처리과정을 거친다. 전역 움직임 추정 및 보상과정을 통해 객체와 배경의 움직임을 분리한다. 이 과정은 제안하는 알고리즘에서 가장 높은 복잡도를 차지하므로, 영상 전체가 아닌 관심영역 단위로 수행함으로써 알고리즘의 고속화를 얻는다^[12]. ST-MRF 모델을 기반으로 객체의 형태와 위치가 예측된다.

Khatoonabadi의 방법에서 나타나는 과분할 및 오차전파 문제를 해결하기 위해 후처리 과정으로 객체 형태 조정(object shape adjustment) 알고리즘이 수행된다. 이 알고리즘은 객체의 경계를 나타내는 변환계수 정보를 활용하여 예측된 객체의 형태를 실제 형태와 유사하게 일치시켜준다. 제안하는 객체 추적 방법의 전체 흐름도는 그림 2와 같다.

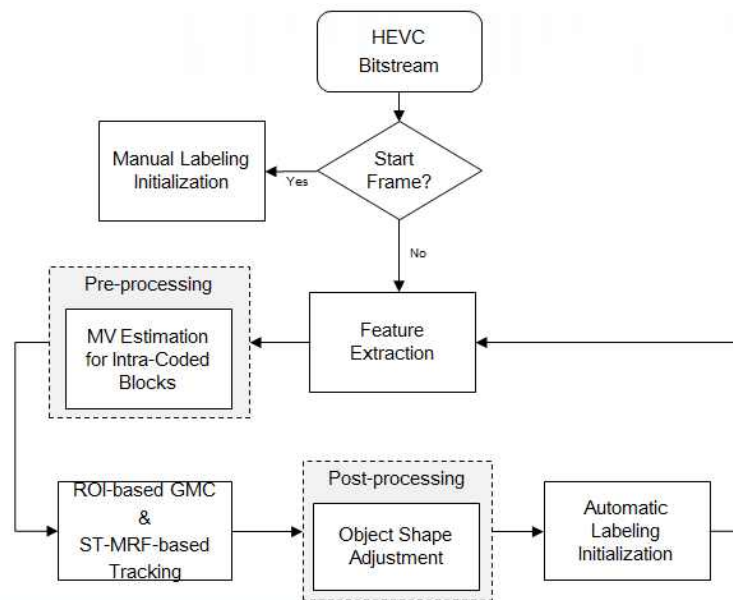


그림 2. 제안하는 객체 추적 방법의 전체 흐름도
Fig. 2. Flow chart of the proposed object tracking

1. ST-MRF 모델 기반 추적 모델

ST-MRF 모델은 강체(rigid object)의 특성을 기반으로 한다^[11]. 객체 추적은 ST-MRF 모델의 마르코프 임의 장 해법을 추론해내는 문제로 생각할 수 있다. 구체적으로, 프레임 임을 작은 블록(4x4 블록)으로 나누어 객체를 포함하는 블록과 아닌 블록을 라벨링(labeling)을 통해 구별한다. $t-1$ 번째 프레임의 블록 라벨 $\omega^{t-1} \in \{0,1\}$ 과 움직임 및 부호화 정보 $\kappa^t = \{\nu^t, o^t\}$ 가 주어졌을 때, t 번째 프레임의 블록 라벨 ω^t 를 추론함으로써 객체 추적이 이루어진다. 여기서 ν^t 는 움직임 벡터를 의미하고 o^t 는 블록의 부호화 모드 및 부호화 단위의 크기를 나타낸다. 최적의 라벨 ω^t 를 찾아내는 방법은 사후확률(posterior probability)을 최대화 하는 것이다. 사후확률 $P(\omega^t | \omega^{t-1}, \kappa^t)$ 는 inter-frame likelihood $P(\omega^{t-1} | \omega^t, \kappa^t)$, intra-frame likelihood $P(\kappa^t | \omega^t)$, 선험적 확률(a priori probability) $P(\omega^t)$ 를 이용한 베이시안 프레임 워크(Bayesian framework)로 식 (1)과 같이 나타낼 수 있다.

$$P(\omega^t | \omega^{t-1}, \kappa^t) = \frac{P(\omega^t | \omega^{t-1}, \kappa^t) \cdot P(\kappa^t | \omega^t) \cdot P(\omega^t)}{P(\omega^{t-1}, \kappa^t)} \quad (1)$$

분모는 ω^t 에 종속이 아니므로 분자를 최대화함으로써 ω^t 에 대한 최대사후확률 해법을 식 (2)와 같이 구할 수 있다.

$$\omega^t = \operatorname{argmax}\{P(\omega^{t-1} | \psi, \kappa^t) \cdot P(\kappa^t | \psi) \cdot P(\psi)\}, \psi \in \Omega \quad (2)$$

여기서 Ω 는 t 번째 프레임의 모든 가능한 레이블을 나타낸다. 식 (2)는 식 (3)과 같이 나타낼 수 있다.

$$\omega^t = \operatorname{argmin}\{-\log P(\omega^{t-1} | \psi, \kappa^t) - \log P(\kappa^t | \psi) - \log P(\psi)\}, \psi \in \Omega \quad (3)$$

또한, Hammersley-Clifford theorem^[14]에 따라 식 (3)의 확률들을 $e^{-E(x)}/Z$ 의 형태를 가지는 깁스(Gibbs)분포를 통해 식 (4~6)과 같이 나타낼 수 있다. $E(x)$ 는 에너지 함수

(energy function)를, Z 는 정규화 상수(normalizing constant)를 나타낸다.

$$P(\omega^{t-1} | \psi, \kappa^t) = \frac{1}{Z_1} \exp\left\{-\frac{1}{\lambda_\Gamma} E_\Gamma(\psi; \omega^{t-1}, \kappa^t)\right\} \quad (4)$$

$$P(\kappa^t | \psi) = \frac{1}{Z_2} \exp\left\{-\frac{1}{\lambda_A} E_A(\psi; \kappa^t)\right\} \quad (5)$$

$$P(\psi) = \frac{1}{Z_3} \exp\left\{-\frac{1}{\lambda_\Phi} E_\Phi(\psi)\right\} \quad (6)$$

식 (4)~(6)에서 세 에너지 함수 E_Γ , E_A , E_Φ 는 각각 시간적 연속성(temporal continuity), 움직임 일관성(motion coherence), 밀집도(compactness)를 나타낸다. λ_Γ , λ_A , λ_Φ 는 계수 인자 (scaling factor)이다. 각각의 에너지 함수 E_Γ , E_A , E_Φ 는 객체를 포함하는 블록들의 각각의 에너지 ε_Γ , ε_A , ε_Φ 의 합으로 나타낼 수 있다.

시간적 연속성은 물체로 판명된 현재 블록을 움직임 벡터를 이용해 이전 프레임으로 사영시킴으로써 구해진다. 현재 블록을 사영시킨 위치에서 현재 블록의 라벨과 이전 블록의 라벨의 겹침의 정도(degree of overlap)가 ε_Γ 가 된다. 움직임 일관성은 일반적으로 강체에 속한 움직임 벡터들은 서로 유사하다는 특성에 기반한다. 객체의 경계부분이나 flat-texture부분에 해당하는 움직임 벡터는 아웃라이어(outlier)에 해당하기 때문에 아웃라이어 제거 알고리즘이 필요하다. 본 논문에서는 Modified Trimmed Mean^[11]방법을 통해 아웃라이어를 제거 한다. 그 후, 객체로 판명된 여러 블록들의 움직임 벡터들을 이용하여 객체의 움직임을 대표하는 대표 움직임 벡터를 PVM (Polar Vector Median)^[11]방법을 통해 구한다. 대표 움직임 벡터와 각각의 움직임 벡터와의 표준편차를 계산하여 움직임 일관성 ε_A 을 구한다. 밀집도는 일반적인 강체들은 조밀한(compact) 형태를 지닌다는 특성에 기반한다. 주변 블록들이 한 객체에 속한다면 현재 블록도 같은 객체에 속할 확률이 높다. 따라서 ε_Φ 을 현재 블록의 8-adjacency에 해당하는 주변 블록들의 라벨들의 가중합(weighted sum)으로 구한다.

식 (7)의 해를 구하기 위해 Stochastic Relaxation(SR)^[15]과 ICM^[13] 알고리즘을 적용할 수 있다. SR이 ICM에 비해 높은 정확도를 보여주지만 계산량이 많다는 단점이 있다. 본 논문에서는 추적 알고리즘의 고속화를 고려하므로 복잡한 SR이 아닌 ICM 알고리즘을 이용한다. ICM을 적용하기 전에 현재 블록들의 라벨들을 초기화해야 한다. 이전 프레임의 라벨들을 전역 움직임 파라미터를 이용해 현재 프레임으로 사영시켜 현재 블록의 라벨을 초기화 시킨다. 그 후, ICM을 통해 블록들의 라벨들은 식 (7)에 정의된 에너지 함수를 감소시키는 방향으로 갱신된다. 이 과정은 반복적으로 수행되며 더 이상 에너지가 감소되지 않거나 정해진 반복 횟수보다 많은 반복이 이루어지는 경우 종료된다. 최종적인 라벨 $\omega^{t-1} \in \{0,1\}$ 가 결정되면 현재 프레임 내에 존재하는 객체의 추적이 완료된다. 위 과정을 프레임단위로 수행하게 되면 결과적으로 동영상에서 객체의 추적이 가능케 된다.

2. 관심영역 기반 전역 움직임 보상

카메라의 움직임으로 발생하는 전역 움직임은 프레임 내의 모든 화소에 영향을 미친다. 전역 움직임은 객체의 순수한 움직임에 추가되기 때문에 정확한 객체의 추적을 위해선 움직임 벡터로부터 전역 움직임을 제거하는 것이 필수적이다. M-estimator^[16]기반 방식은 전역 움직임을 나타내기 위해 6개의 파라미터를 사용하는 어파인(Affine) 모델을 사용한다^[17]. 주어진 전역 움직임 파라미터 $\mathbf{m} = [m_1, \dots, m_6]$ 에 의해 (x, y) 에 위치하던 블록은 (x', y') 위치로 이동하게 되고, (x', y') 위치는 식 (8)과 같이 나타낼 수 있다. 움직임의 정도는 움직임 벡터 $\mathbf{v}(x, y)$ 로 표시하며 식 (9)와 같이 나타낼 수 있다.

$$x' = m_1 + m_2x + m_3y, \quad y' = m_4 + m_5x + m_6y \quad (8)$$

$$\mathbf{v}(x, y) = (x' - x, y' - y) \quad (9)$$

$$\omega^t = \operatorname{argmin} \left\{ \frac{1}{\lambda_\Gamma} \cdot \sum_{n:\psi(n)=1} \varepsilon_\Gamma(n; \omega^{t-1}, \boldsymbol{\kappa}^t) + \frac{1}{\lambda_A} \cdot \sum_{n:\psi(n)=1} \varepsilon_A(n; \boldsymbol{\kappa}^t) \right\}, \psi \in \Omega \quad (7)$$

\mathbf{m} 과 영상 전체에 존재하는 블록들의 각각의 위치 및 움직임 벡터를 이용하여 전역 움직임 변환을 식 (10)과 같은 형태로 표현할 수 있으며 식 (10)을 풀어쓰면 식 (11)과 같다.

$$\mathbf{V} = \mathbf{H} \cdot \mathbf{m} \quad (10)$$

$$\begin{bmatrix} v_x^{(1)} + x^{(1)} \\ v_y^{(1)} + y^{(1)} \\ \vdots \\ v_x^{(N)} + x^{(N)} \\ v_y^{(N)} + y^{(N)} \end{bmatrix} = \begin{bmatrix} 1 & x^{(1)} & y^{(1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x^{(1)} & y^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x^{(N)} & y^{(N)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x^{(N)} & y^{(N)} \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{bmatrix} \quad (11)$$

식 (11)에서 $(x^{(n)}, y^{(n)})$ 는 n 번째 위치한 블록의 중심 좌표를 의미하고, $v_x^{(n)}, v_y^{(n)}$ 는 해당 블록이 가지는 움직임 벡터 $\mathbf{v}(x, y)$ 의 x, y 요소를 각각 의미한다. \mathbf{m} 은 최소자승법 해법(least squares solution)을 통해 식 (12)와 같이 구할 수 있다.

$$\mathbf{m} = (\mathbf{H}^T \cdot \mathbf{W} \cdot \mathbf{H})^{-1} \cdot \mathbf{H}^T \cdot \mathbf{W} \cdot \mathbf{V} \quad (12)$$

최종 수식을 보면 \mathbf{W} 행렬이 추가된 것을 볼 수 있다. 이는 전체 영상에 존재하는 아웃라이어 움직임 벡터를 제거하기 위한 행렬로 각각의 움직임 벡터에 대한 가중치로 구성되어 있다. 부호화기에서의 움직임 벡터는 실제 해당 블록의 움직임을 반영하지 않고, 부호화 효율이 가장 높은 위치를 나타낸다. 따라서, 블록의 움직임과 상이한 움직임 벡터가 존재하게 되고, 이러한 아웃라이어들은 전역 움직임 추정 과정에 악영향을 미치게 된다. Arvanitidou에 따르면 8x8 이상의 큰 블록들은 배경에 속할 확률이 높지만 작은 블록들은 객체에 속할 확률이 높다^[16]. 따라서, 큰 크기를 가지는 블록들을 전역 움직임 추정에 사용하고, 나머지는 작은 블록들은 제외한다. 또한, 이전 프레임에서 객체로 판명

된 블록들은 전역 움직임과 상이한 움직임을 보일 확률이 높기 때문에 전역 움직임 추정 과정에서 제외시킨다. \mathbf{W} 행렬은 식 (13)과 같이 구성된다.

$$\mathbf{W} = \begin{bmatrix} w^{(1)} & 0 & \dots & \dots & \dots \\ 0 & w^{(1)} & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & 0 & w^{(N)} & 0 \\ 0 & \dots & \dots & 0 & w^{(N)} \end{bmatrix}, \quad (13)$$

$$w^{(n)} = \begin{cases} \left(1 - \frac{\varepsilon_{(n)}^2}{(c\mu)^2}\right)^2 & \varepsilon_{(n)} < c\mu \\ 0 & \varepsilon_{(n)} \geq c\mu \end{cases}$$

$$\mu = \frac{1}{M} \sum_m \varepsilon^{(n)}, \quad \varepsilon^{(n)} = |v_x^{(n)} - \tilde{v}_x^{(n)}| + |v_y^{(n)} - \tilde{v}_y^{(n)}| \quad (14)$$

$w^{(n)}$ 는 n 번째 위치에 존재하는 블록에 대한 가중치를 나타내며 식 (13)과 같이 나타낸다. c 는 상수를 나타낸다. $\varepsilon^{(n)}$ 는 n 번째 위치에 존재하는 블록의 실제 움직임 벡터 $\mathbf{v}(x, y)$ 와 \mathbf{m} 에 의해 예측된 움직임 벡터 $\tilde{\mathbf{v}}(x, y)$ 간의 오차를 의미하며 식 (14)와 같이 나타낸다. $\tilde{v}_x^{(n)}, \tilde{v}_y^{(n)}$ 는 움직임 벡터 $\tilde{\mathbf{v}}(x, y)$ 의 x, y 요소를 각각 의미하고. M 은 움직임 벡터의 수이다. M-estimator는 반복적 알고리즘(iterative algorithm)을 통해 해를 구한다. 매 반복마다 위에서 정의한 \mathbf{W} 행렬을 갱신하면서 가장 적은 오차를 갖는 \mathbf{m} 을 유추해

낸다.

제안하는 방법에서 가장 높은 복잡도를 차지하는 모듈은 전역 움직임 보상이다. 전역 움직임 보상 알고리즘은 영상 전체를 대상으로 수행되는 연산으로 처리해야 할 데이터의 양이 많고, 행렬 연산 및 역행렬 연산 등을 포함하고 있어 복잡도가 높다. 본 논문에서는 관심영역 기반 전역 움직임 보상 알고리즘을 적용하여 추적 알고리즘을 고속화한다^[12]. 그림 3은 전역 움직임 보상을 영상 전체에 적용한 경우와 관심영역에 적용한 경우를 보여준다. 그림 3 (c)를 보면 관심영역 내부만 전역 움직임 보상이 적용된 것을 볼 수 있으며, 배경과 객체의 움직임을 분리시킬 수 있다.

3. 화면 내 예측 블록에 대한 움직임 벡터 추정

화면 내 예측 모드로 부호화된 블록은 움직임 정보가 없으므로 해당 블록에 대한 ST-MRF를 수행할 수 없다. 따라서 본 논문에서는 PVM^[11] 방법을 통해 화면 내 예측 모드로 부호화된 블록의 움직임 벡터를 예측한다. PVM은 화면 내 예측 블록의 4-neighborhood에 위치하는 블록들의 움직임 벡터들로부터 예측된다. 주변 블록의 움직임 벡터를 4x4 단위로 참조하여 벡터 리스트 V 를 구성한다. 그림 4와 같은 경우를 생각해 보자. 그림 4의 경우 벡터 리스트 V 는 식 (15)와 같이 구성할 수 있다.

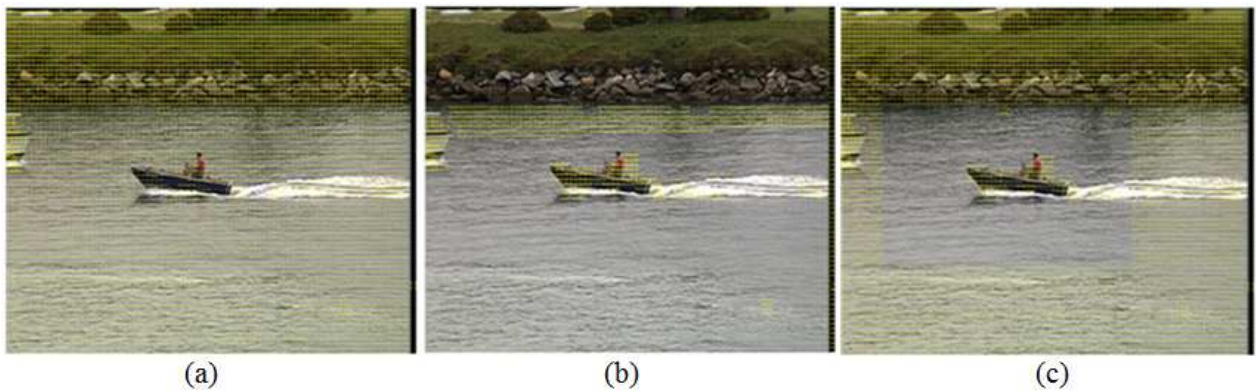


그림 3. 전역 움직임 보상. (a) 전역 움직임 보상 전 움직임벡터. (b) 전체영역 기반 전역 움직임 보상. (c) 관심영역 기반 전역 움직임 보상
Fig. 3. GMC. (a) Motion vector before GMC. (b) Motion vector after GMC. (c) Motion vector after ROI-based GMC

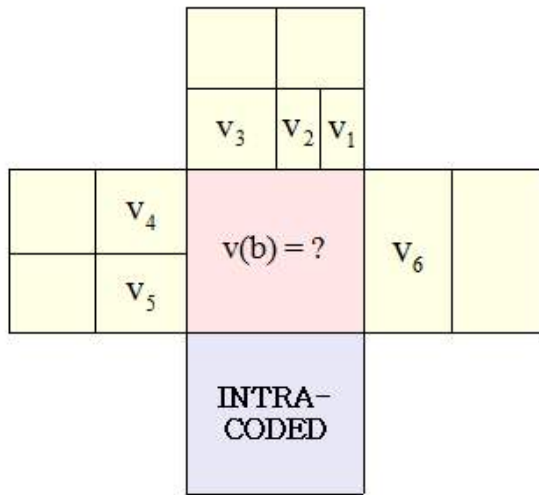


그림 4. 화면 내 예측 모드로 부호화된 블록에 대한 움직임 벡터 할당
Fig. 4. Motion vector assignment for intra-coded block

$$V = (v_1, v_2, v_3, v_3, v_4, v_4, v_5, v_5, v_6, v_6, v_6, v_6) \quad (15)$$

4x4 단위로 움직임 벡터를 참조하므로 v_3, v_4, v_5 의 경우 두 번, v_6 의 경우 네 번의 동일한 움직임 벡터가 참조되었다. 화면 내 예측 블록의 대표 움직임 벡터를 계산하기 위해 극좌표(polar coordinate)가 사용된다. n 개의 움직임 벡터를 가지는 벡터 리스트 $V_p = (v_i)_{i=1:n}$ 에서 $-\pi$ 부터 π 의 각도에 따라 움직임 벡터들을 정렬시킨다. 식 (16)을 만족

하는 I 에 대하여 $m = \lfloor (n+1)/2 \rfloor$ 개의 움직임 벡터로 구성된 V_p 를 구성하며 V_p 는 식 (17)과 같다.

$$I = \operatorname{argmin} \sum_{i=j}^{j+m-2} \theta_i \quad (16)$$

$$V_p = (v_i)_{i=I:I+m-1} \quad (17)$$

여기서 θ_i 는 움직임 벡터 v_i 와 v_{i+1} 사이의 각도를 나타낸다. PVM v_p 의 각도와 크기는 식 (18)~(19)으로 결정된다.

$$\angle v_p = \operatorname{median}(\angle v_i)_{i=I:I+m-1} \quad (18)$$

$$\|v_p\|_2 = \operatorname{median}(\|v_i\|_2)_{i=I:I+m-1} \quad (19)$$

그림 5는 PVM의 한 예를 보여준다. 구해진 PVM v_p 는 화면 내 예측 블록의 모든 4x4블록의 움직임 벡터로 할당된다.

4. 변환계수 기반 객체형태 조정 알고리즘

Khatoonabadi의 방법에서는 그림 6과 같이 배경과 객체가 비슷한 움직임 벡터를 가지는 경우에 과분할 결과를 초

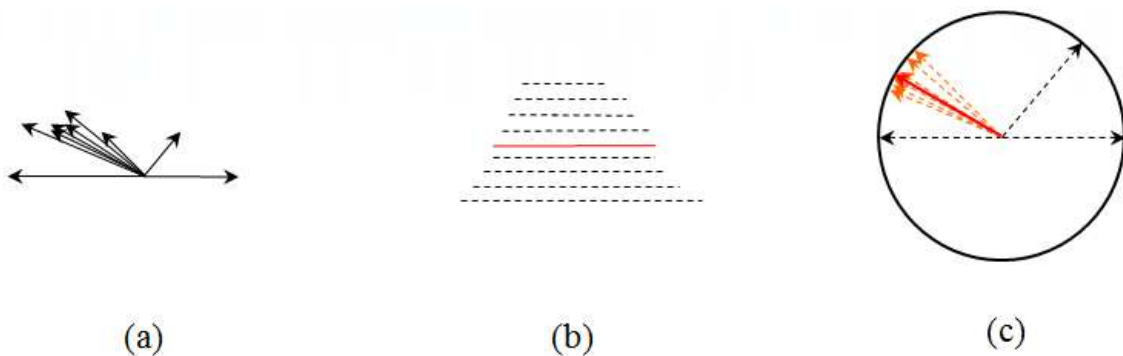


그림 5. PVM. (a) 입력 벡터. (b) 빨간 라인 : PVM의 크기. (c) 입력 벡터의 각도. 빨간 벡터 : PVM의 각도.
Fig. 5. PVM. (a) Input vectors. (b) Lengths of input vectors. Red line : representative length. (c) Angles of vectors. Red vector : representative angle

래하는 문제가 있다. 움직임 벡터는 객체의 순수한 움직임에 따라 결정되는 것이 아니라 부호화 효율 측면에서 최적의 위치를 가리키게 된다. 배경과 객체를 서로 다른 블록으로 부호화 하는 것보다 하나의 블록으로 부호화하는 것이 효율이 좋을 수 있다. 이 경우, 배경과 객체가 동일한 움직임 벡터를 갖게 되어 객체로 오분류 되는 경우가 발생한다. 과분할 문제가 발생하면 연속된 영상에서 오분류된 객체 정보를 참조하므로 오차전파 현상이 발생해 전체적인 추적 성능이 저하된다. 객체 형태 조정 알고리즘은 이 문제를 해결하기 위해 고안되었다. 이 알고리즘은 변환계수를 특징점으로 사용해 객체의 경계를 식별하고, 예측된 객체의 경계면과의 유사성을 기반으로 예측된 객체의 경계를 최적의 위치로 이동시킨다. 객체의 형태를 임의로 바꿔주는 것이 아니므로 연속된 영상에서 객체의 형태변화가 거의 없다는 가정을 기반으로 한다.

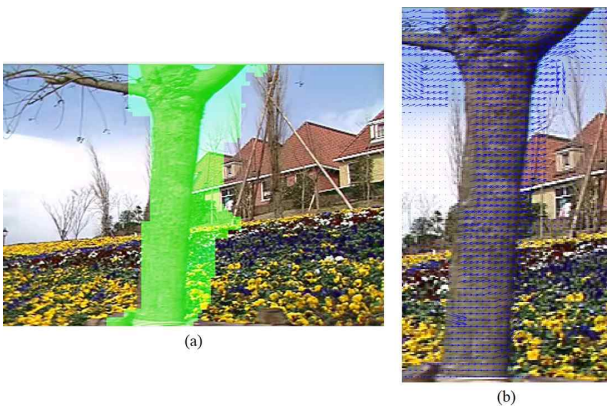


그림 6. 과분할 결과. (a) 추적 결과. (b) 움직임 벡터.
 Fig. 6. Result of over-segmentation. (a) Tracking result. (b) Motion vectors

객체 형태 조정 알고리즘은 실제 객체의 경계를 나타내는 특징점으로 변환계수의 에너지를 이용한다. 특징점 추출 과정에서 변환계수의 이진화를 수행한다. 변환계수의 에너지는 화소단위로 정보가 존재하므로 4x4 블록 단위로 절댓값 평균을 취하여 다른 특징점들과 해상도를 맞춘다. 그 후, 아웃라이어(outlier)를 제거하기 위해 2x2 미디언 필터를 적용하고, log 값을 취해 전체적인 값들의 dy-

namic range를 조정한다. 그 후, 특정 문턱값 이상의 값을 가지는 변환계수들만 이진화하여 객체의 경계를 나타내는 특징점으로 선택된다. 문턱값은 실험을 통해 3.0으로 정하였다.

그림 7은 변환계수를 특징점으로 활용한 객체 형태 조정 알고리즘의 흐름도를 나타낸다. 경계 추출 과정에서는 예측된 객체의 경계를 상, 하, 좌, 우 4개의 면으로 나누어 추출한다. 경계는 예측된 객체의 각각의 행과 열을 따라 블록 위치의 최소값과 최대값을 통해 추출한다. 경계 추출 과정이 완료되면 경계 위치 예측과정에서 각각의 경계면을 중심으로 검색 영역을 설정한 후, 검색 영역 안에서 예측된 경계의 형태와 특징점들이 이루는 경계의 형태의 유사성이 가장 높은 위치를 구한다. 유사성을 나타내는 기준으로 연관성(correlation)을 사용한다. 연관성이 가장 높은 위치가 결정되면 경계 조정 과정을 통해 해당 위치로 예측된 경계를 조정한다.

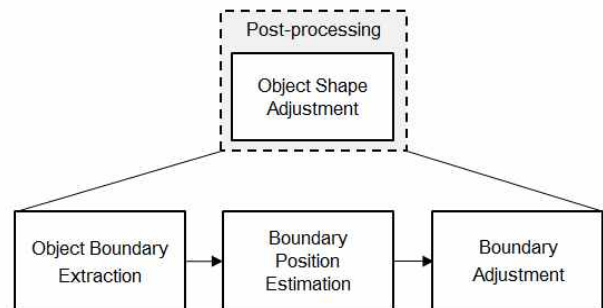


그림 7. 객체형태 조정 알고리즘의 흐름도
 Fig. 7. Flow chart of object shape adjustment algorithm

그림 8은 객체 형태 조정 알고리즘의 적용 과정을 순차적으로 나타낸다. 그림 8의 (a)는 추적기를 통해 예측된 객체로 배경과 객체의 움직임이 비슷하여 과분할된 결과를 볼 수 있다. (b)는 객체의 좌우 경계를 나타낸다. (c)에서 추출된 경계를 중심으로 탐색 영역을 설정하고, 변환계수가 이루는 경계와의 유사성이 최대가 되는 위치를 탐색한다. (d)에서 탐색된 위치로 경계를 이동시켜줌으로써 과분할 및 오차전파 문제를 방지한다.

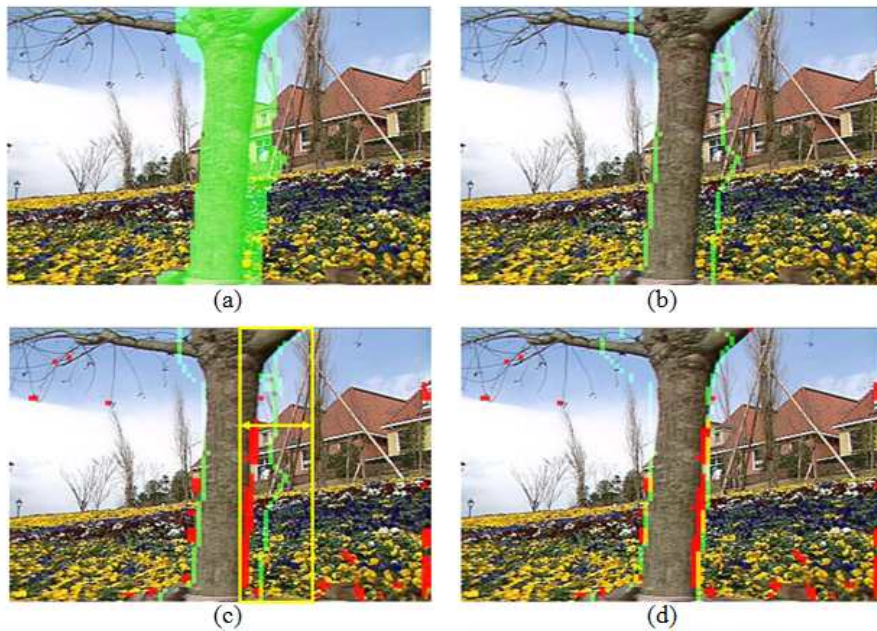


그림 8. 객체형태 조정 알고리즘의 수행과정. (a) 예측된 객체. (b) 경계면 추출. (c) 변환계수와의 유사성 탐색. (d) 경계면 조정
 Fig. 8. Shape adjustment algorithm process. (a) Object estimation. (b) Object boundary extraction. (c) Search a correlation with transform coefficient. (d) Boundary adjustment

IV. 실험 및 결과

1. 실험 환경

제안하는 방법의 성능을 평가하기 위해 Intel i7-2600 3.40GHz 프로세서, 14.0GB 메모리를 사용하였고 Microsoft Visual Studio 2010에서 C++를 기반으로 알고리즘을 구현하였다. 실험에 사용된 영상은 MPEG의 SIF (352x240), CIF (352x288) 크기의 표준영상을 사용하였다. 각각의 영상들은 HEVC test Model (HM) 14.0버전으로 부호화 되었으며 low-delay P configuration의 Common Test Conditions (CTC)를 이용하였다. 참조영상의 수는 1로 설정하였으며, Quantization Parameter (QP)는 22로 하였다. 복호화기를 통해 부호화된 비트스트림을 복호화 하면서 객체 추적에 필요한 움직임 벡터, 예측 단위 정보, 변환계수를 추출하였다. ST-MRF 모델의 계수인자 λ_r , λ_A , λ_ϕ 값은 각각 1, 2/3, 0.25로 Khatoonabadi의 방법[11]과 동일한 값을 사용하였다. 대부분의 영상들이 좌, 우의 움직임을 많이 갖게

되고 이로 인해 좌, 우 경계에서 과분할 및 오차전파 문제를 보이게 된다. 따라서 이번 실험에서는 형태 조정 알고리즘을 좌, 우 경계만을 고려해 적용하였다.

추적성능 분석에는 ground truth 정보가 활용되었다. 추적결과와 ground truth를 비교하였을 때 객체로 판정한 부분과 ground truth의 객체가 일치한 픽셀의 수를 true positive로, 객체로 판정한 부분이 ground truth의 배경인 픽셀의 수를 false positive로, 배경으로 판정한 부분이 ground truth의 객체인 픽셀의 수를 false negative로 나타내었다. 추적성능을 정확도(precision), 재현율 (recall), F-measure로 비교하였는데, 각 값들은 식 (20)과 같이 정의되며, TP는 true positive 값, FP는 false positive 값, FN은 false negative 값을 의미한다.

$$\begin{aligned}
 Precision &= \frac{TP}{TP+FP} \\
 Recall &= \frac{TP}{TP+FN} \\
 F\text{-measure} &= \frac{2 \times Precision \times Recall}{Precision + Recall}
 \end{aligned} \tag{20}$$

2. 실험 결과

제안하는 방법과 Khatoonabadi의 방법의 추적성능은 표 1과 같다. 제안하는 방법이 Khatoonabadi의 방법보다 정확도 측면에서 6.1% 더 좋은 성능을 보이는 것을 알 수 있다. 정확도가 높다는 것은 객체가 있다고 판정했을 경우 실제 객체가 있을 확률이 높다는 것이다. 반면, 재현율은 5% 정도 감소하였다. 이는 객체가 있을 경우 객체로 판정할 확률이 상대적으로 낮다는 것을 의미한다. 본 논문의 형태 조정

알고리즘에서 실제 객체의 경계보다 조정된 경계가 실제 객체 안쪽에 위치하는 경우가 생겨 재현율이 낮아지는 결과를 보였다. 정확도와 재현율의 종합적인 성능을 나타내는 F-measure가 0.2% 향상된 성능을 보이며 과분할 및 오차전과 문제가 발생하는 FlowerGarden 영상에서 큰 성능 향상을 보인다.

그림 9는 FlowerGarden 영상에 대한 추적결과를 나타낸다. Khatoonabadi의 방법의 경우 추적영역이 오른쪽으로 점점 퍼지는 과분할 문제가 나타난다. 제안하는 방법은 형

표 1. Khatoonabadi의 방법과의 추적성능 비교

Table 1. Tracking performance comparison of the proposed method with respect to the Khatoonabadi's method

Sequence	Frame	Precision (%)		Recall (%)		F-measure (%)	
		Pro	Ref	Pro	Ref	Pro	Ref
City (CIF)	100	91.68	94.5	95.39	95.4	93.46	94.9
FlowerGarden (SIF)	50	76.49	59.3	84.77	94.0	80.15	71.2
TableTennis (SIF)	25	91.80	96.3	92.14	89.0	91.91	92.4
Stefan (CIF)	90	95.72	85.4	44.79	59.2	60.10	68.7
HallMonitor (CIF)	60	89.30	70.9	68.82	80.9	77.60	75.4
MobileCalendar (CIF)	100	84.05	77.8	82.54	83.1	83.00	80.1
Coastguard (CIF)	100	64.41	62.8	89.75	90.1	74.61	73.4
Foreman (CIF)	100	97.86	95.5	80.03	85.9	87.62	90.2
Average		86.41	80.3	79.78	84.7	81.06	80.8

* Pro는 제안하는 추적방법을, Ref는 Khatoonabadi의 방법을 나타낸다.

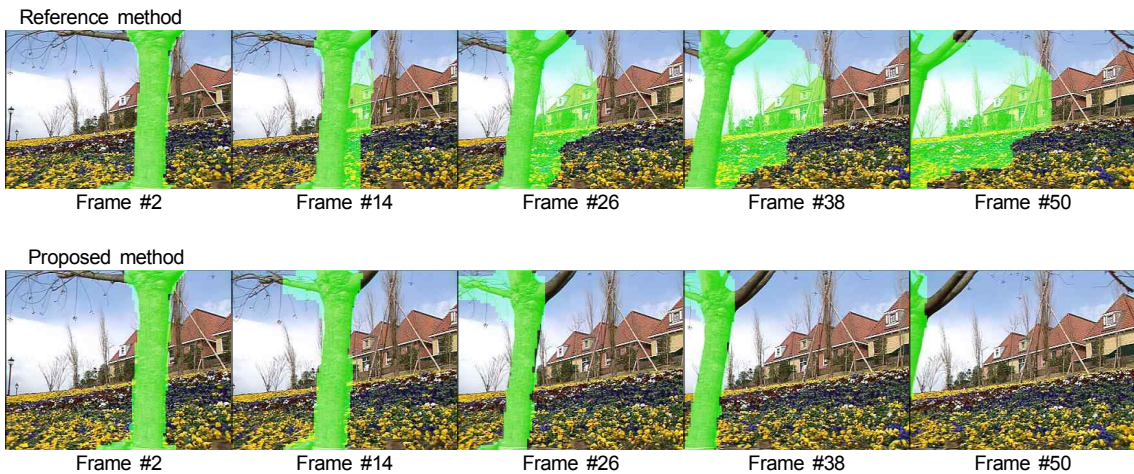


그림 9. FlowerGarden 영상에 대한 객체 추적 결과

Fig. 9. Tracking results for FlowerGarden

태 조정 알고리즘을 통해 이 현상을 효과적으로 제거하였다.

그림 10은 MobileCalendar 영상 200프레임에 대한 추적 결과를 객체를 중심으로 확대해 나타낸다. Khatoonabadi의 방법은 추적 대상인 공을 추적하다가 공과 기차가 부딪힌 후, 기차로 추적영역이 커지는 문제를 보인다. 또한, 시간이 지남에 따라 결국 공의 추적을 실패하고 기차를 추적하게 되는 결과를 초래한다. 이는 공과 기차가 부딪히면서 인접

한 영역에 비슷한 움직임 벡터를 갖게 되어 같은 객체로 오분할 되기 때문이다. 오분할이 된 객체는 연속된 프레임에서 오차전과 문제를 야기시켜 추적영역을 크게 만든다. 제안하는 방법은 추적대상인 공을 기차에 대한 오분할 없이 추적하는 결과를 보인다. 이는 공과 기차가 부딪혔을 때, 인접한 블록들이 비슷한 움직임 벡터를 가지지만, 공과 기차의 영상 특성이 달라 경계부분에서 큰 변환계수를 가지게 되어 형태 조정 알고리즘이 오분할 문제를 효과적으로

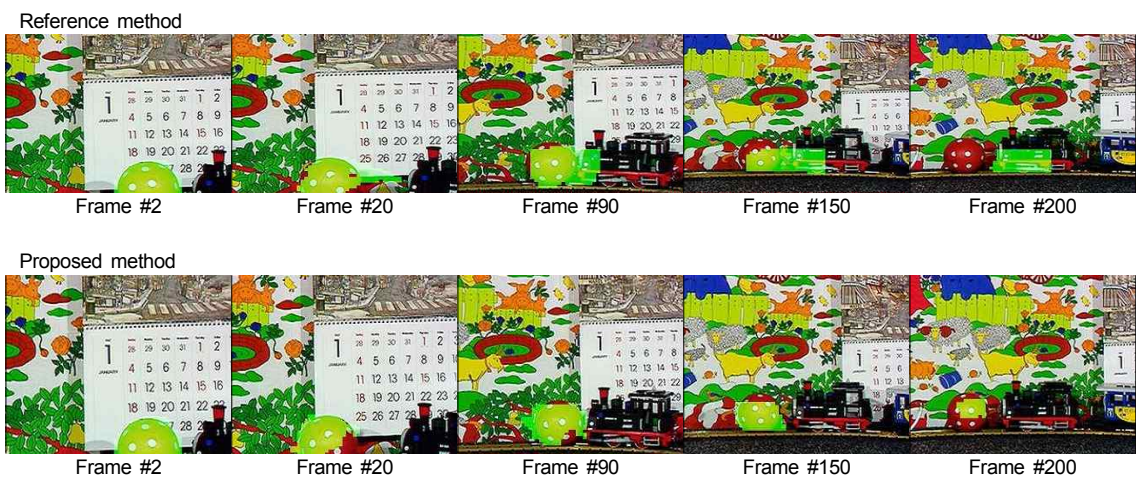


그림 10. MobileCalendar 영상에 대한 객체 추적 결과
Fig. 10. Tracking results for MobileCalendar

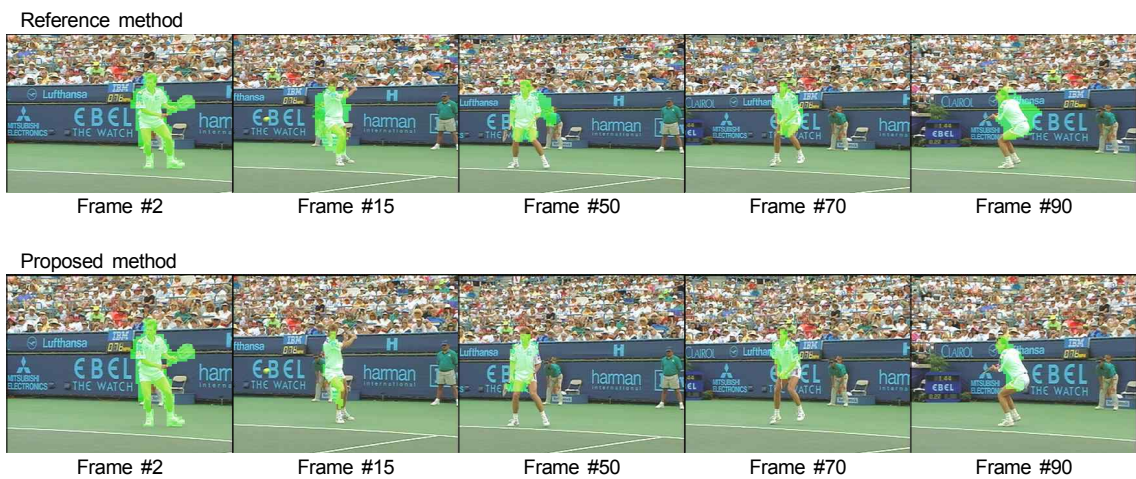


그림 11. Stefan 영상에 대한 객체 추적 결과
Fig. 11. Tracking results for Stefan

방지하였기 때문이다.

그림 11은 Stefan 영상에 대한 추적결과를 나타낸다. 사람의 팔과 다리는 몸통에 비해 움직임이 많고 변화가 심하기 때문에 두 추적방법 모두 팔과 다리의 추적이 잘 이루어지지 않는다. 이로 인해 Stefan 영상의 평균 재현율은 다른 영상에 비해 크게 감소하게 된다. 그림 12는 Stefan 영상에 대한 프레임별 추적성능을 나타낸다. Khatoonabadi의 방법의 경우 15번째와 50번째 프레임 주변에서 과분할 문제가 따라 정확도가 크게 감소함을 확인할 수 있다. 제안하는 방법은 형태 조정 알고리즘을 통해 이와 같은 과분할 문제를 해결하여 정확도가 상승하였다. 하지만 추적 결과가 실제 객체보다 작아지면서 재현율은 크게 감소하게 되고 재현율이 감소함에 따라 F-measure 역시 감소하게 된다. 이러한 현상은 변환계수의 주된 에너지 분포가 실제 경

계보다 안쪽에 위치함에 따라 예측된 경계면도 안쪽으로 조정되었기 때문이다. 대체적으로 변환계수의 주된 에너지 분포는 실제 경계 또는 실제 경계보다 안쪽에 위치하는 경향을 보인다. 이로 인해 추적결과가 실제 객체보다 작아지는 경우가 발생하게 되고 재현율이 감소하는 경향을 보인다.

한 프레임에 대한 제안하는 방법의 수행시간은 표 2에 제시하였다. Khatoonabadi의 방법에 비하여 평균적으로 약 10배의 고속화를 얻었지만 이는 MATLAB 기반으로 구현된 Khatoonabadi의 방법을 C++를 기반으로 구현해 최적화함으로써 얻어진 속도향상을 포함한 수치다. 이를 제외한 알고리즘 측면의 속도향상은 약 2배로 관심영역 기반 전역 움직임 보상과정으로 얻어진 결과이다.

부호화기의 특성상 대부분의 블록들은 화면 내 예측 모

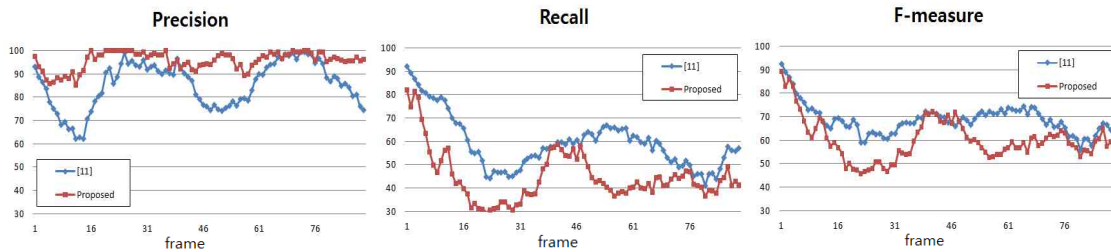


그림 12. Stefan 영상에 대한 추적성능
 Fig. 12. Tracking performance for Stefan

표 2. 한 프레임에 대한 모듈별 수행시간
 Table 2. Processing time of each module per frame

Sequence	Pre-processing (ms)		GMC (ms)		ST-MRF tracking (ms)		Post-processing (ms)		Total (ms)	
	Pro	Ref	Pro	Ref	Pro	Ref	Pro	Ref	Pro	Ref
City (CIF)	0.01	0.5	7.2	52.9	1.9	7.3	0.3	0	9.4	60.7
FlowerGarden (SIF)	0.01	0.4	5.3	45.5	1.3	7.0	0.2	0	6.9	52.9
TableTennis (SIF)	0.02	0.5	3.3	38.5	1.6	6.2	0.3	0	5.2	45.2
Stefan (CIF)	0.04	1.3	4.1	50.8	0.6	3.3	0.1	0	4.8	55.4
HallMonitor (CIF)	0.04	1.8	1.9	45.9	0.6	3.6	0.1	0	2.7	51.2
MobileCalendar (CIF)	0.01	0.5	2.6	49.5	0.4	2.7	0.1	0	3.2	52.7
Coastguard (CIF)	0.01	0.6	2.8	48.6	0.4	2.7	0.2	0	3.4	51.9
Foreman (CIF)	0.04	1.5	4.9	48.1	2.7	9.3	0.3	0	8.0	58.9
Average	0.02	0.9	4.0	47.5	1.2	5.3	0.2	0	5.4	53.6

* Pro는 제안하는 방법을, Ref는 Khatoonabadi의 방법을 나타낸다.

드가 아닌 화면 간 예측 모드로 부호화된다. 따라서 화면 내 예측모드로 부호화된 블록에 대해서만 적용되는 전처리 과정은 차지하는 복잡도가 낮다. 관심영역 기반 전역 움직임 보상 과정은 많은 반복연산과 행렬연산을 요구하기 때문에 가장 높은 복잡도를 차지하고 있다. ST-MRF 추적과정은 객체가 존재할 가능성이 높은 관심영역 내에서만 연산이 수행된다. 따라서, 비교적 낮은 복잡도를 보이며 객체의 크기에 따라 복잡도가 달라진다. 객체의 크기가 큰 Foreman 영상보다 객체의 크기가 작은 Coastguard 영상에서 수행속도가 6배 이상 빠르다. 후처리과정에 해당하는 형태 조정 알고리즘의 복잡도가 작은 이유는 이미 예측된 객체의 경계를 식별하고 경계에 대해서만 연산을 수행하고 반복연산을 요구하지 않기 때문이다.

V. 결 론

본 논문에서는 최근 표준화가 완료된 HEVC 표준을 따르는 비트스트림 상에서의 객체 추적 방법을 제안하였다. 화소 정보를 얻기 위한 복호화를 수행하지 않고, 비트스트림에 존재하는 정보들을 추출해 이용한다. 제안하는 방법은 ST-MRF 모델을 기반으로 하며, 기존방법에서 나타나는 과분할 문제를 해결하고 추가적인 고속화를 얻는다. HEVC 비트스트림에서 움직임을 반영하는 3가지 특징점을 규정하였다. 움직임 벡터는 객체의 움직임을, 예측 단위의 크기와 변환 계수의 에너지는 객체의 경계를 나타낸다. 제안하는 방법은 화면 내 예측 모드로 부호화된 블록들은 움직임 벡터가 없으므로 주변 블록들의 움직임 벡터를 이용한 PVM을 통해 움직임 벡터를 할당해준다. 전역 움직임 보상 과정을 통해 카메라의 움직임 등으로 발생하는 전역 움직임을 제거하여 객체의 순수한 움직임만을 추출해 객체와 배경을 구분하며 이를 영상 전체가 아닌 관심영역 단위로 수행함으로써 알고리즘을 고속화한다. ST-MRF 모델을 통해 객체의 공간-시간적 특성을 반영하여 객체의 위치와 형태가 예측된다. 이전 프레임의 객체는 현재 프레임에 연속적으로 나타날 확률이 높다는 특성을 시간적 연속성으로, 객체 내의 움직임 벡터들은 서로 동일한 특성을 보인다는

것을 움직임 일관성으로, 일반적인 객체는 조밀한 형태를 지닌다는 특성을 밀집도로 모델링하여 최대사후확률 해법에 적용한다. 기존방법에서 나타나는 과분할 및 오차전과 문제를 해결하기 위해 형태 조정 알고리즘이 적용된다. 이 알고리즘은 객체의 경계를 따라 높은 변환계수의 에너지가 존재하는 특성을 특징점으로 사용한다. 이 특징점을 기반으로 현재 예측된 객체의 경계를 특징점들이 이루는 경계와의 유사성을 기반으로 조정시켜준다.

제안하는 방법은 기존방법보다 추적성능 및 복잡도 측면에서 향상된 성능을 보인다. 평균적으로 정확도가 6.1% 향상하고 재현율은 5.0% 하락하면서 종합적인 성능을 평가하는 F-measure는 0.2% 향상하였지만 FlowerGarden 영상에서 F-measure 값은 최대 9% 정도 향상되었다. 실험영상 중 과분할 및 오차전과 문제가 두드러지는 MobileCalendar 영상에서 기존방법과 다르게 목표물이 다른 객체와 충돌한 후 분리되는 상황에서도 목표물을 놓치지 않고 지속적으로 추적할 수 있었다.

참 고 문 헌 (References)

- [1] ITU-T, "Advanced Video Coding for Generic Audiovisual Services," Rec. H.264/ISO IEC 14996-10 AVC, 2003.
- [2] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, Y.-K. Wang, and T. Wiegand, "High Efficiency Video Coding (HEVC) Text Specification Draft 10," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, JCTVC-L1003, Geneva, CH, Jan. 2013.
- [3] V. Mezaris, I. Kompatsiaris, N. V. Boulgouris, and M. G. Strintzis, "Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 5, pp. 606 - 621, May 2004.
- [4] C. Käs and H. Nicolas, "An approach to trajectory estimation of moving objects in the H.264 compressed domain," in Proc. 3rd Pacific Rim Symp. Adv. Image Video Technol., pp. 318 - 329, 2009.
- [5] W. Fei and S. Zhu, "Mean shift clustering-based moving object segmentation in the H.264 compressed domain," IET Image Process., vol. 4, no. 1, pp. 11 - 18, Feb. 2010.
- [6] S. Treetasanavorn, U. Rauschenbach, J. Heuer, and A. Kaup, "Bayesian method for motion segmentation and tracking in compressed videos," in Proc. 27th DAGM Conf. Pattern Recognit., pp. 277 - 284, 2005.
- [7] S. Treetasanavorn, U. Rauschenbach, J. Heuer, and A. Kaup, "Stochastic motion coherency analysis for motion vector field segmentation on compressed video sequences," in Proc. IEEE Workshop

- Image Anal. Multimedia Interact. Services, pp. 1 - 4, Apr. 2005.
- [8] W. Zeng, J. Du, W. Gao, and Q. Huang, "Robust moving object segmentation on H.264/AVC compressed video using the block-based MRF model," Real-Time Imaging, vol. 11, no. 4, pp. 290 - 299, Aug. 2005.
- [9] Y. M. Chen, and I. V. Bajic, "Moving region segmentation from compressed video using global motion estimation and markov random fields," IEEE Transactions on Multimedia, vol. 13, no. 3, June 2011.
- [10] C. M. Mak, and W. K. Cham, "Real-time video object segmentation in H.264 compressed domain," IET image Processing, vol. 3, Iss. 5, Oct, 2009.
- [11] S. H. Khatoonabadi and I. V. Bajic, "Video object tracking in the compressed domain using spatio-temporal Markov Random Fields," IEEE Transactions on Image Processing, vol. 22, no. 1, pp.300-313, Jan. 2013.
- [12] D. M. Park, D. K. Lee, S. M. Kim, and S. J. Oh "Fast ST-MRF tracking using ROI-based GMC," The Korean Society Of Broadcast Engineers, 2014. Nov.
- [13] J. Besag, "On the statistical analysis of dirty pictures," J. Royal Stat. Soc. B, vol. 48, no. 3, pp. 259 - 302, 1986.
- [14] J. E. Besag, "Spatial interaction and the statistical analysis of lattice systems," J. Royal Stat. Soc., Ser. B, vol. 36, no. 2, pp. 192 - 236, 1974.
- [15] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 6, no. 6, pp. 721 - 741, Nov. 1984.
- [16] M. G. Arvanitidou, A. Glantz, A. Krutz, T. Sikora, M. Mrak, and A. Kondoz, "Global motion estimation using variable block sizes and its application to object segmentation," in Proc. IEEE Workshop Image Anal. Multimedia Interact. Services, London, U.K., pp. 173 - 176, May 2009.
- [17] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. 2nd ed, Cambridge Univ. Press, Cambridge, U.K., pp. 39 - 44, 2004.

저 자 소 개



박 동 민

- 2014년 2월 : 광운대학교 전자공학과 학사
- 2014년 3월 ~ 현재 : 광운대학교 전자공학과 석사과정
- ORCID:<http://orcid.org/0000-0002-7063-4354>
- 주관심분야 : 영상압축, 컴퓨터 비전



이 동 규

- 2012년 2월 : 광운대학교 전자공학과 학사
- 2014년 2월 : 광운대학교 전자공학과 석사
- 2014년 3월 ~ 현재 : 광운대학교 전자공학과 박사과정
- ORCID:<http://orcid.org/0000-0003-0713-854X>
- 주관심분야 : 영상압축, 컴퓨터 비전



오 승 준

- 1980년 2월 : 서울대학교 전자공학과 학사
- 1982년 2월 : 서울대학교 전자공학과 석사
- 1988년 5월 : 미국 Syracuse University 전기/컴퓨터공학과 박사
- 1982년 3월 ~ 1992년 8월 : 한국전자통신연구원 멀티미디어연구실 실장
- 1986년 7월 ~ 1986년 8월 : NSF Supercomputer Center 초청 학생연구원
- 1987년 5월 ~ 1988년 5월 : Northeast Parallel Architecture Center 학생연구원
- 1992년 3월 ~ 1992년 8월 : 충남대학교 컴퓨터공학부 겸임교수
- 1992년 9월 ~ 현재 : 광운대학교 전자공학과 교수
- 2002년 3월 ~ 현재 : SC29-Korea 의장 및 MPEG Forum 부의장
- ORCID:<http://orcid.org/0000-0002-5036-3761>
- 주관심분야 : 비디오 데이터 처리, 영상압축, 멀티미디어시스템