

특집논문 (Special Paper)

방송공학회논문지 제21권 제6호, 2016년 11월 (JBE Vol. 21, No. 6, November 2016)

<http://dx.doi.org/10.5909/JBE.2016.21.6.913>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

## 다중 랜덤 워커를 이용한 객체 추적 기법

문 주 혁<sup>a)</sup>, 김 한 울<sup>a)</sup>, 김 창 수<sup>a)†</sup>

## Visual Object Tracking by Using Multiple Random Walkers

Juhyeok Mun<sup>a)</sup>, Han-Ul Kim<sup>a)</sup>, and Chang-Su Kim<sup>a)†</sup>

### 요 약

본 논문에서는 다중 랜덤 워커(multiple random walkers)에 기반한 객체 추적 기법을 제안한다. 우선 서포트 벡터 머신(support vector machine)을 이용한 분류기 기반 객체 추적 기법을 소개한다. 다음으로 영상의 영역에 대한 특징 벡터 중 배경으로부터 추출된 특징 벡터를 억제하는 기법을 제안한다. 영역에서 배경 요소를 찾기 위해 다중 랜덤 워커를 이용한 전경 및 배경 추출 방법을 제시한다. 배경 요소가 억제된 특징 벡터를 이용하여 학습된 서포트 벡터 머신은 객체와 배경이 유사한 영상, 객체가 다른 물체에 의해 가려지는 영상 등에서 객체와 배경을 확실하게 구분하고, 추적 알고리즘은 정확한 객체 추적을 수행한다. 또한, 객체 추적 알고리즘의 응용에서 중요한 속도 문제를 크게 개선하는 방법을 제안한다. 마지막으로 실험을 통해 제안하는 기법이 높은 처리 속도를 유지하면서 동시에 기존 기법보다 우수한 추적 성능을 보임을 확인한다.

### Abstract

In this paper, we propose the visual tracking algorithm that takes advantage of multiple random walkers. We first show the tracking method based on support vector machine as [1] and suggest a method that suppresses feature vectors extracted from backgrounds while preserve features vectors from foregrounds. We also show how to discriminate between foregrounds and backgrounds. Learned by reducing influences of backgrounds, support vector machine can clearly distinguish foregrounds and backgrounds from the image whose target objects are similar to backgrounds and occluded by another object. Thus, the algorithm can track target objects well. Furthermore, we introduce a simple method improving tracking speed. Finally, experiments validate that proposed algorithm yield better performance than the state-of-the-art trackers on the widely-used benchmark dataset with high speed.

Keyword : Computer Vision, Tracking, Random walkers

a) 고려대학교 전기전자전파공학부(School of Electrical, Korea University)

† Corresponding Author : 김창수(Chang-Su Kim)

E-mail: cskim@mcl.korea.ac.kr

Tel: +82-2-3290-3806

ORCID: <http://orcid.org/0000-0002-4276-1831>

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2015R1A2A1A10055037)

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음 (No. IITP-2016-R2720-16-0007)

※ 이 논문의 연구결과 중 일부는 “2016년 한국방송·미디어공학회 하계학술대회”에서 발표한 바 있음.

· Manuscript received September 19, 2016; Revised October 25, 2016; Accepted October 25, 2016.

## I. 서론

객체 추적은 동영상에서 목표 객체의 초기 위치로부터 이후 프레임에서의 위치들을 자동으로 추정하는 기술로 다양한 연구가 활발하게 진행되고 있다<sup>[1,2,3,7,8]</sup>. 최근에는 기계 학습<sup>[9]</sup>에 기법을 통해 객체와 배경을 효과적으로 판별할 수 있는 객체 모델을 학습하는 분류기 기반의 추적 기법들이 객체 추적 알고리즘의 성능을 크게 향상시켰다.

분류기는 영상의 영역으로부터 추출된 특징 벡터를 이용하여 객체와 배경 간의 차이를 학습한다. 일반적으로 객체 추적에서 학습 샘플의 후보 영역은 박스 형태로 주어진다. 하지만 객체의 모양은 다양하므로 분류기 학습에 제공되는 샘플의 특징 벡터에는 배경 정보가 포함될 수 있다. 배경 정보로 학습된 분류기는 객체와 배경을 잘 구분하지 못하여 분류기 기반 추적 알고리즘의 성능은 현저히 떨어진다.

본 논문에서는 이러한 문제점을 해결하기 위해 다중 랜덤 워커(multiple random walkers) 기법<sup>[4]</sup>을 이용한다. 객체 박스를 그래프로 표현하기 위해 다수의 부분 영역으로 나누고, 각 부분 영역을 노드로 하는 다중 랜덤 워커 시뮬레이션을 수행한다. 이로부터 전경과 배경에 해당하는 요소를 추출하고 객체가 존재할 확률에 대한 가중치를 계산한다. 이 가중치를 특징 벡터에 곱해주어 분류기 학습에서 배경 요소가 미치는 영향을 최소화한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 기법에 대해 설명한다. 3장에서는 실험을 통해 제안 기법이 우수함을 보이고, 마지막으로 4장에서는 본 논문의 결론을 맺는다.

## II. 제안하는 기법

### 1. 분류기 기반 객체 추적

본 논문에서는 영역을 기술하기 위해, [3]의 기법과 같이 영역의 색상, 경계와 지역적 특성을 활용하여 특징벡터를 구성한다. 먼저 영역의 지역적 특성을 반영하기 위해 영역을 64개의 겹치지 않는 블록으로 분할한다. 각 블록은 24차원의 색상 히스토그램과, 8차원의 경계 히스토그램으로 표

현한다. 전체 영역은 각 블록의 특징 벡터를 연결하여 구성한 2048차원의 특징 벡터로 기술한다.

본 논문에서 목표 객체를 분류기<sup>[1]</sup>으로 모델링 한다. 학습된 서포트 벡터 머신과 이전 프레임에서의 객체 위치를 중심으로 추출한 후보 영역으로부터 현재 프레임에서의 객체 위치를 서포트 벡터 머신의 점수를 최대화하는 후보 영역으로 결정한다. 목표 객체의 위치가 결정되면 다음과 같은 식(1)의 구조화 서포트 벡터 머신의 목적함수를 확률적 경사 하강법 알고리즘<sup>[6]</sup>을 사용하여 최적화 하고, 식(2)과 같이 분류기를 갱신한다.

$$\mathbf{w}^* = \min_{\mathbf{w}} \|\mathbf{w}\|^2 + C \sum_{n=1}^l \max_{y \in Y} (\Delta(y_n, y) + \langle \mathbf{w}, \psi(\mathbf{x}_n, y) \rangle - \langle \mathbf{w}, \psi(\mathbf{x}_n, y_n) \rangle) \quad (1)$$

$$\mathbf{w}_t = \alpha \mathbf{w}_{t-1} + (1 - \alpha) \mathbf{w}^* \quad (2)$$

$\mathbf{x}$ 와  $y$ 는 샘플 영상과 객체 박스 라벨을,  $\Delta$ 는 두 라벨 간 거리를 나타낸다.  $\langle \cdot \rangle$ 는 내적,  $\psi(\cdot)$ 는 영상의 특징 벡터를 구하는 함수이다. 본 논문에서  $\alpha$ 는 0.7로 고정하였다.

### 2. 다중 랜덤 워커

박스 형태의 후보 영역을 특징 벡터로 표현하는 방식에는 영역 안에 배경 블록을 포함 할 수 있기 때문에 2,048차원의 특징 벡터 역시 배경 정보를 포함하는 단점을 가진다. 목표 객체 추적을 위해 분류기는 배경으로부터 객체를 분류해내는 작업을 수행하기 때문에, 배경으로부터 얻어진 특징 벡터는 분류기를 오염시켜 추적 알고리즘의 성능을 저하시킨다. 따라서 높은 추적 성능을 달성하기 위해서는 특징 벡터 구성 과정에서 배경 정보에 대한 영향을 억제할 필요가 있다.

본 절에서는 다중 랜덤 워커 시뮬레이션을 통해 객체 영역에 포함된 배경 정보를 억제하는 방법을 제안한다. 먼저, 랜덤 워커 시뮬레이션을 수행하기 위해 객체 영역 안 64개의 블록을 노드로 하는 그래프를 구성한다. 각 블록의 특징 벡터를 이용하여 노드 간 유사도를 계산하고, 노드 간 유사도를 바탕으로 친화도 행렬  $\mathbf{A}$ 를 구한다. 랜덤 워커 시뮬레

이선에서 워커는 친화도 행렬  $\mathbf{A}$ 에 의해 노드를 이동하며, 그래프 전체에서 확률 값을 통해 분포를 나타낸다. 친화도 행렬  $\mathbf{A}$ 는 다음과 같이 구한다.

$$w_{ij} = \exp\left(-\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{\sigma^2}\right), \quad (3)$$

$$a_{ij} = \frac{w_{ij}}{\sum_k w_{kj}}, \quad (4)$$

$$\mathbf{A} = [a_{ij}]. \quad (5)$$

$i, j$ 는 각각 노드를 나타내는 첨자이며  $\mathbf{x}_i, \mathbf{x}_j$ 는 각각  $i, j$  노드 블록을 표현하는 특징 벡터이다.

제안하는 기법에서는 전경과 배경을 명확하게 구분하기 위해 영상의 전경과 배경에 해당하는 두 워커에 대해 랜덤 워커 기법을 동시 적용하는 다중 랜덤 워커 기법<sup>[4]</sup>을 이용한다. 다중 랜덤 워커 기법은 전경 노드와 배경 노드의 분포를 상호 고려하여 전경과 배경이 반발 작용을 통해 그 경계를 더 강하게 나타내는 결과를 보인다. 식 (6)과 (7)는 각각 전경과 배경의 랜덤 워커 시뮬레이션을 나타낸다.

$$\mathbf{p}_f^{(t+1)} = (1 - \epsilon)\mathbf{A}\mathbf{p}_f^{(t)} + \epsilon\mathbf{r}_f^{(t)}, \quad (6)$$

$$\mathbf{p}_b^{(t+1)} = (1 - \epsilon)\mathbf{A}\mathbf{p}_b^{(t)} + \epsilon\mathbf{r}_b^{(t)}, \quad (7)$$

다중 랜덤 워커 기법은 전경 노드 분포  $\mathbf{p}_f^{(t)}$ 와 배경 노드

분포  $\mathbf{p}_b^{(t)}$ 를 활용하여 재시작 분포  $\mathbf{r}$ 을 설정한다. 재시작 분포는 랜덤 워커 시뮬레이션에서, 일정 확률로 노드를 정해진 위치로 이동시켜 알고리즘의 강인함을 높인다. 본 논문에서는 전경과 배경의 반발 작용을 나타내기 위해 전경 노드 분포와 배경 노드 분포를 이용하여 전경 확률이 높은 노드는 배경 확률이 낮게, 배경 확률이 높은 노드는 전경 확률이 낮게 나오도록 워커 간 상호관계를 설정하였다. 구체적으로 재시작 확률  $\mathbf{r}^{(t)}$ 은 다음과 같이 구한다.

$$\mathbf{r}_k^{(t)} = (1 - \delta^t)\mathbf{r}_k^{(t-1)} + \delta^t\Phi_k^{(t)}(\mathbf{P}^{(t)}), \quad \mathbf{P}^{(t)} = \{\mathbf{p}_k^{(t)}\}_{k=1}^K, \quad (8)$$

$$\mathbf{r}_k^{(t)} = \mathbf{r}_k^{(t-1)}, \quad (9)$$

$$\mathbf{r}_k^{(t)} = \Phi_k^{(t)}(\mathbf{P}^{(t)}). \quad (10)$$

$$\mathbf{p}_k^{(t)} = [p^{(t)}(\mathbf{x}_1|\omega_k), \dots, p^{(t)}(\mathbf{x}_N|\omega_k)]^T, \quad (11)$$

$$p^{(t)}(\omega_k|\mathbf{x}_i) = \frac{p^{(t)}(\mathbf{x}_i|\omega_k)p^{(t)}(\omega_k)}{\sum_l p^{(t)}(\mathbf{x}_i|\omega_l)p^{(t)}(\omega_l)}, \quad (12)$$

$\Phi_k^{(t)}$ 는 재시작 규칙 함수로, 매 시각  $t$ 에서 모든 워커가 갖는 확률 분포로부터 새로운 재시작 분포를 산출한다.  $k$ 는 전경과 배경을 의미하는 첨자이다. 식 (8)의  $\delta$ 는 0과 1사이의 값을 갖는 실험 상수로  $\delta = 0$ 의 경우, 시각  $t$ 가 변해도 재시작 분포  $\mathbf{r}_k^{(t)}$ 은 식 (9)와 같이 시간 불변성을 갖고, 기존의

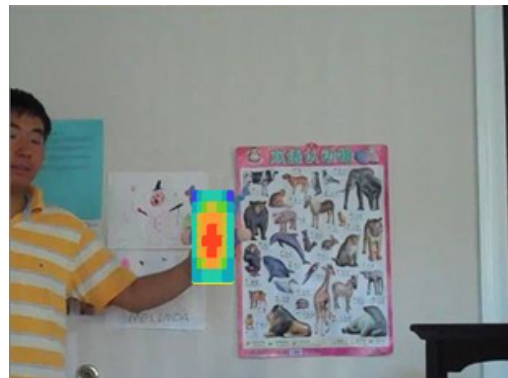
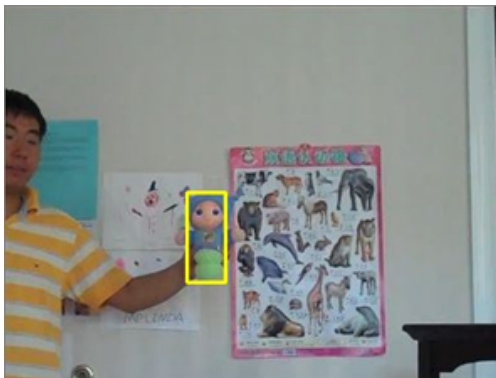


그림 1. 부분 영역별 가중치 예시: 적색에 가까울수록 객체를 포함할 확률이 높다

Fig. 1. An example of patch weights. Red depicts a high patch weight, while blue depicts a low patch weight

RWR(Random Walker with Restart)와 동일하게 동작한다. 반면  $\delta = 1$ 의 경우 시각  $t$ 에서의 재시작 분포  $\mathbf{r}_k^{(t)}$ 은 이전 시각  $t-1$ 에서의 재시작 분포와는 독립적으로 구해진다. 본 논문에서는  $\delta$ 를 0.9로 고정하였다.  $p^{(t)}(x_i|\omega_k)$ 는 워커  $k$ 가 노드  $i$ 에 분포할 확률을 나타낸다. 베이지 정리를 이용하여 노드  $i$ 가 워커  $k$ 에 의해 차지될 확률  $p^{(t)}(\omega_k|x_i)$ 은 식 (12)과 같이 구한다. 두 조건부 확률을 곱한 가능도(likelihood)를 바탕으로 전경과 배경의 반발을 표현한다.

$$\phi_{k,i}^{(t)} = \alpha \cdot p^{(t)}(\omega_k|x_i) \cdot p^{(t)}(x_i|\omega_k), \quad (13)$$

$$\Phi_k^{(t)}(\mathbf{P}^{(t)}) = \alpha \mathbf{Q}_k^{(t)} \mathbf{p}_k^{(t)}. \quad (14)$$

$\Phi_k^{(t)}$ 는 각 대각 요소가  $\phi_{k,i}^{(t)}$ 인 대각 행렬이다.  $\alpha$ 는  $\Phi_k^{(t)}$ 를 확률 분포로 만들어주기 위한 정규화 상수이다. 식 (6)과 식 (7)를 일정  $T$ 만큼 시행하여 얻어진 전경 확률과 배경 확률을 얻는다. 전경과 배경 확률을 이용하여 각 블록에서 객체가 존재할 확률은 식 (15)과 같이 구한다.

$$\Psi^{(t)} = \frac{1}{1 + e^{c(\mathbf{p}_i^{(t)} - \mathbf{p}_b^{(t)})}} \quad (15)$$

$c$ 는 실험적으로 정한 상수로 본 논문에서는  $c$ 를 0.7로 고정하였다. 블록에서 객체가 존재할 확률은 특징 벡터에 가중치로 곱해져 배경의 영향을 억제한다. 그림 1은 식 (10)으로 얻어진 가중치를 도시한다.

### 3. 알고리즘 고속화

객체 추적 알고리즘의 응용에서 자동 주행, 감시 시스템 등과 같이, 입력은 실시간 카메라 영상이 두드러진다. 실시간 카메라 영상을 처리하기 위해서는 알고리즘이 고속으로 동작할 필요가 있다.

본 논문에서 제안하는 알고리즘은 하나의 후보 영역을 64개의 패치로 나누어 모든 패치에 대해 특징 벡터를 추출하기 때문에 후보 영역 하나에 많은 연산을 요구한다. 따라서 알고리즘의 속도는 후보 영역의 개수에 민감하다. 기존의 알고리즘은 탐색 영역 내에서 슬라이딩 윈도우 기법을 이용하여 많은 후보 영역을 다루기 때문에 알고리즘의 속도가 매우 느리다.

본 논문에서는 객체 추적에서 인접한 프레임 간 객체의 위치가 크게 변하지 않는 사실에 주목하였고, 이전 프레임에서 객체를 옳게 찾았다면, 현재 프레임에서 객체의 위치는 이전 프레임과 가까이 위치한다고 가정하였다. 따라서 이전 프레임의 객체 위치를 기준으로 탐색 영역을 설정하고 중심에서 거리가 먼 영역과 가까운 영역으로 나누었다. 식 (16)은 전체 탐색 영역의 반경을 나타내고 식 (17)은 중심과 가까운 영역의 반경을 나타낸다. 중심과 가까운 영역에서는 슬라이딩 윈도우 기법을 적용하였고, 이외의 영역에서는 일정한 간격을 두고 후보 영역을 추출하여 전체 후보 영역의 수를 줄였다. 본 논문에서는 20% 미만의 후보 영역만으로 객체를 추적하여 기존 대비 4배 이상의 속도



그림 2. (a) 기존 기법<sup>[1,3]</sup>에서의 탐색 영역 (b) 제안하는 탐색 영역

Fig. 2. (a) Search region of previous algorithm<sup>[1,3]</sup> (b) Proposed search region

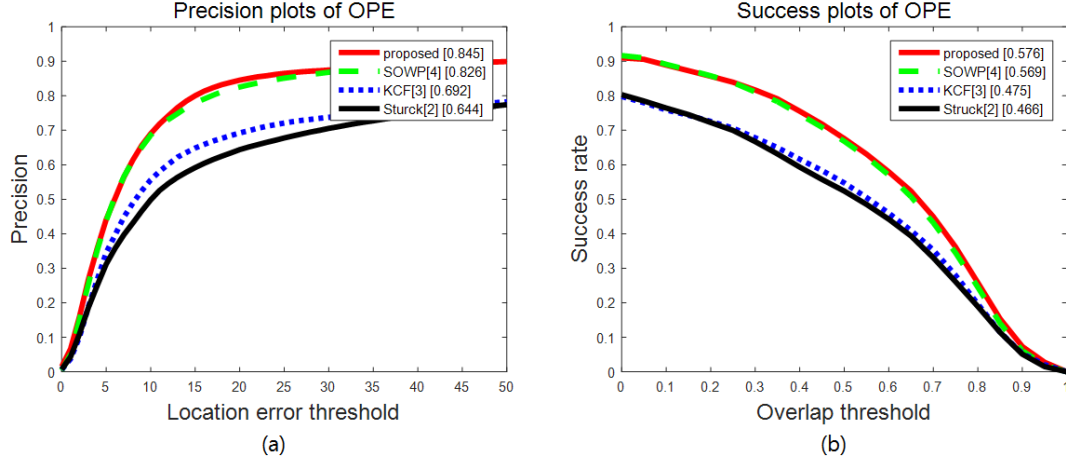


그림 3. TB-100 벤치마크에 대한 전체 성능 비교: (a) 정밀도 (b) 성공률  
Fig. 3. Overall performances on the TB-100: (a) Precision (b) Success rate

향상을 확인하였다.

$$r = \sqrt{w \cdot h}, \quad (16)$$

$$r_{\text{coarse}} = \frac{\sqrt{w \cdot h}}{2}, \quad (17)$$

$$s = \left\lfloor \frac{w}{8} \right\rfloor. \quad (18)$$

$r$ 은 탐색영역의 반경,  $w, h$ 는 각각 객체 박스의 너비와 높이를 나타낸다. 전체 탐색 영역은 기존 탐색 영역의 크기와 동일하다. 그림 2(b)의 파란색 영역은 이전 프레임에서의 객체의 위치와 거리가 멀기 때문에 객체가 위치할 확률이 적다고 판단하여, 슬라이딩 윈도우 기법을 1픽셀 간격이 아닌 식 (18)의  $s$  픽셀 간격마다 후보 영역을 추출한다. 반경  $r_{\text{coarse}}$ 로 표현되는 작은 탐색 영역은 이전 프레임의 객체와 가까운 영역이므로, 객체가 위치할 확률이 높다고 가정하여 1픽셀 간격으로 후보 영역을 추출한다. 이렇게 추출된 후보 영역의 수는 기존 후보 영역의 수보다 20% 미만으로, 전체 알고리즘 속도가 약 4배 이상 향상되었다.

### III. 실험 결과

본 논문에서는 알고리즘의 정량적 평가를 위해 TB-100

벤치마크 영상<sup>[5]</sup>에 대해 기존 기법<sup>[1,2,3]</sup>들과 제안기법을 실험하고 비교하였다. 실험은 Intel i7 2.6GHz CPU, 16GB RAM의 PC에서 이루어졌다. 성능 지표는 객체 추적 분야에서 자주 사용되는 평가 지표인 정밀도(precision)와 성공률(success rate)을 이용하였다. 정밀도는 추정한 객체 위치와 실제 위치의 차가 20 픽셀 이내인 프레임의 비율을 의미하며 성공률은 추정한 영역과 실제 영역 간의 교집합 넓이가 합집합 넓이의 절반 이상인 프레임의 비율을 나타낸다.

그림 3는 최근 객체 추적 벤치마크<sup>[5]</sup>에서 우수한 성능을 보인 기존 기법<sup>[1,2,3]</sup>들과 제안 기법을 비교한다. 정밀도와 성공률 모두에서 제안하는 기법이 더 높은 성능을 보이고 있다.

그림 4는 객체 추적을 방해하는 요인을 갖는 동영상에 대해 제안하는 기법의 추적 결과를 도시한다. 그림 4(a)와 4(b)의 동영상은 객체와 비슷한 외형의 배경이 있어 추적이 어렵다. 그림 4(c)와 4(d)의 동영상은 객체가 배경에 의해 일부 가려지는 경우를 보여주며, 그림 4(e)와 4(f)의 동영상은 안개, 플래시 등으로 객체가 완전히 가려지거나 심한 색 왜곡을 보이는 경우이다. 다양한 방해 요인에도 불구하고 본 논문의 알고리즘은 대부분의 동영상에서 목표 객체 추적에 성공한다. 다만, 그림 4(e)와 같이 오랜 시간 객체가 사라지는 영상에서는 분류기의 오염이 회복 불가능한 정도에 이르러 추적이 실패하는 모습을 보인다. 이는 향후 연구



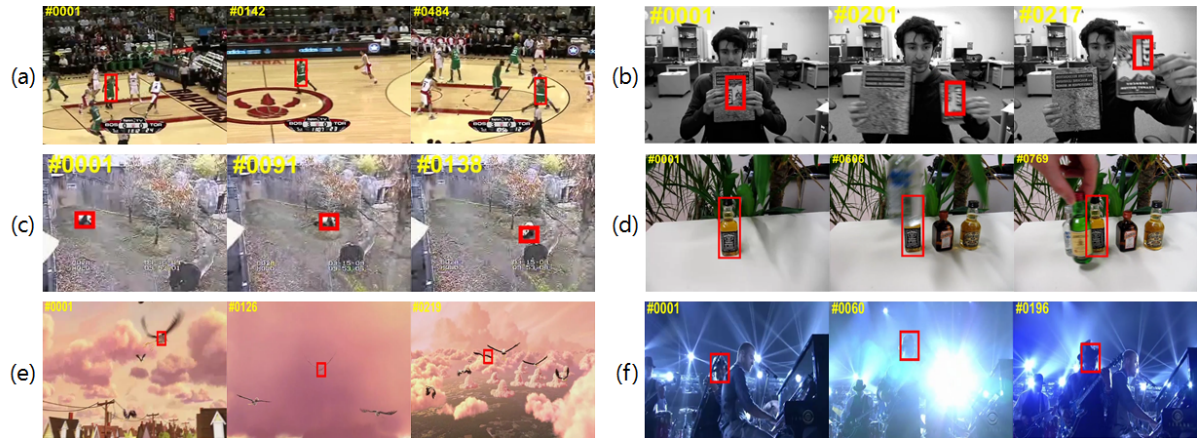


그림 4. TB-100 벤치마크 동영상에서의 추적 결과 예시. 붉은색 박스는 목표 객체를 표시한다.

Fig. 4. Examples of tracking results on the TB-100 sequences. Red boxes contain target objects. (a) Basketball, (b) Clifbar, (c) Panda, (d) Liquor, (e) Bird1, (f) Shaking

에서 해결해야 할 문제이다.

#### IV. 결 론

본 논문에서는 분류기 기반의 객체 추적 기법에서 배경이 분류기에 미치는 영향을 억제하기 위한 가중치 모델을 제안하였다. 억제할 배경 요소를 추출하기 위해 객체 박스를 다수의 블록으로 나누어 다중 랜덤 워커 시뮬레이션을 수행하였다. 각 블록에 객체가 존재할 확률을 해당 블록의 특징 벡터에 가중치로 곱하여 배경의 영향을 억제하였다. 실험 결과는 제안하는 기법을 통해 객체 추적 벤치마크 영상에서 추적 정확도를 크게 개선함을 보였고 기존 기법과 비교를 통해 제안 기법의 성능이 우수함을 확인하였다.

#### 참 고 문 헌 (References)

[1] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in Proc. ICCV., pp. 263 - 270, 2011.

[2] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583 - 596, Mar. 2015.

[3] H. U. Kim, D. Y. Lee, J. Y. Sim, and C. S. Kim, "SOWP: Spatially Ordered and Weighted Patch Descriptor for Visual Tracking," in Proc. ICCV., pp. 3011 - 3019, 2015.

[4] C. W. Lee, W. D. Jang, J. Y. Sim, and C. S. Kim, "Multiple random walkers and their application to image cosegmentation," in Proc. CVPR, pp. 3837-3845, Jun. 2015.

[5] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1834 - 1848, Sep. 2015.

[6] S. Shwartz, Shai, et al. "Pegasos: Primal estimated sub-gradient solver for svm." Mathematical programming vol. 127, no. 1, pp. 3-30, 2011.

[7] B. J. Choi, B. W. Yoon, J. K. Song, and J. Park, "Implementation of Pedestrian Detection and Tracking with GPU at Night-time," JBE, vol. 20, no. 3, May. 2015.

[8] J. Choi, Y. Choe, and Y. G. Kim, "Histogram Equalization Based Color Space Quantization for the Enhancement of Mean-Shift Tracking Algorithm," JBE, vol. 19, no. 3, May. 2014.

[9] S. Kim, and Y. M. Ro, "A Study for Improved Human Action Recognition using Multi-classifiers," JBE, vol. 19, no. 2, Mar. 2014.

[10] Tong, Hanhang, C. Faloutsos, and J. Y. Pan, "Random walk with restart: fast solutions and applications." Knowledge and Information Systems, vol. 14, no. 3, 2008.

---

— 저 자 소 개 —

---



문 주 혁

- 2016년 2월 : 고려대학교 전기전자전파공학부 학사
- 2016년 ~ 현재 : 고려대학교 전기전자공학과 석사과정
- 주관심분야: 컴퓨터 비전, 딥 러닝



김 한 울

- 2014년 2월 : 고려대학교 전기전자전파공학부 학사
- 2016년 2월 : 고려대학교 전기전자공학과 석사과정
- 2016년 3월 ~ 현재 : 고려대학교 전기전자공학과 박사과정
- 주관심분야: 컴퓨터 비전, 딥 러닝



김 창 수

- 1994년 2월 : 서울대학교 공학사
- 1996년 2월 : 서울대학교 공학석사
- 2000년 8월 : 서울대학교 공학박사
- 2000년 8월 ~ 2001년 12월 : USC 박사후연구원
- 2002년 1월 ~ 2003년 7월 : 서울대학교 박사후연구원
- 2005년 8월 ~ 현재 : 고려대학교 전기전자공학부 교수
- ORCID : <http://orcid.org/0000-0002-4276-1831>
- 주관심분야 : 비디오 및 3D 영상처리, 멀티미디어 통신, 컴퓨터 비전