

특집논문 (Special Paper)

방송공학회논문지 제25권 제2호, 2020년 3월 (JBE Vol. 25, No. 2, March 2020)

<https://doi.org/10.5909/JBE.2020.25.2.192>

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

CPU 환경에서의 실시간 동작을 위한 딥러닝 기반 다중 객체 추적 시스템

김 경 훈^{a)}, 허 준 호^{a)}, 강 석 주^{a)†}

Towards Real-time Multi-object Tracking in CPU Environment

Kyung Hun Kim^{a)}, Jun Ho Heo^{a)}, and Suk-Ju Kang^{a)†}

요 약

최근 딥러닝 모델을 기반으로 한 객체 추적 알고리즘의 활용도가 증가하고 있다. 영상에서의 다중 객체의 추적을 위한 시스템은 대표적으로 객체 검출 알고리즘과 객체 추적 알고리즘의 연쇄된 형태로 구성되어있다. 하지만 여러 모듈로 구성된 연쇄 형태의 시스템은 고성능 컴퓨팅 환경을 요구하며 실제 어플리케이션으로의 적용에 제한사항으로 존재한다. 본 논문에서는 위와 같은 객체 검출-추적의 연쇄 형태의 시스템에서 객체 검출 모듈의 연산 관련 프로세스를 조정하여 저성능 컴퓨팅 환경에서도 실시간 동작을 가능하게 하는 방법을 제안한다.

Abstract

Recently, the utilization of the object tracking algorithm based on the deep learning model is increasing. A system for tracking multiple objects in an image is typically composed of a chain form of an object detection algorithm and an object tracking algorithm. However, chain-type systems composed of several modules require a high performance computing environment and have limitations in their application to actual applications. In this paper, we propose a method that enables real-time operation in low-performance computing environment by adjusting the computational process of object detection module in the object detection-tracking chain type system.

Keyword : multi object tracking, data association, real time object tracking

a) 서강대학교 전자공학과(Dept. of Electronic Engineering, Sogang University)

† Corresponding Author : 강석주(Suk-Ju Kang)

E-mail: sjkang@sogang.ac.kr

Tel: +82-2-705-8466

ORCID: <https://orcid.org/0000-0002-4809-956X>

※ 이 논문의 연구 결과 중 일부는 한국방송·미디어공학회 2019년 “추계학술대회”에서 발표한 바 있음.

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행되었으며 (No. 2018R1D1A1B07048421), 또한 행정안전부 스마트 도로조명 플랫폼 개발 및 실증 연구 개발사업의 연구비지원(과제번호 19PQWO-B153369-01)에 의해 수행되었으며, 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2020-2018-0-01421)

※ This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2018R1D1A1B07048421) and a grant(19PQWO-B153369-01) from Smart road lighting platform development and empirical study on test-bed Program funded by Ministry of the Interior and Safety of Korean government, and the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2018-0-01421) supervised by the IITP(Institute for Information & communications Technology Promotion).

· Manuscript received January 14, 2020; Revised March 3, 2020; Accepted March 3, 2020.

Copyright © 2020 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

I. 서론

비디오 영상에서의 자동 물체 인식 및 추적 기술은 자율 주행^[1], 스포츠 경기 분석^[2], 무인 감시 시스템^[3] 등 다양한 분야에서 활용되고 있으며, 국내외에서 관련 기술의 개발이 계속해서 급증하고 있다. 하지만 기존의 영상 내 여러 객체를 동시에 추적하는 다중 객체 추적(Multi Object Track-ing; MOT) 또는 다중 대상 추적(Multi Target Track-ing; MTT) 기술은 무궁무진한 활용도에 비해 실제 적용 면에서 부족한 성능을 보인다.

먼저 다중 객체 추적의 성능 측면 중 정확도 관련하여 크게 두 가지 어려움이 있는데 하나는 검출기의 성능이 좋지 못해서 검출을 잘못하는 경우와 두 번째로 추적하는 대상이 다른 객체에 가려져 검출이 불가능한 경우에 추적 구현 시 어려움이 있다. 다중 객체 추적 시스템의 구성 측면 상 객체 검출 과정이 선행적으로 필수적이며 이러한 객체의 검출 정보를 토대로 추적을 수행하는 추적 시스템은 검출기의 정확성에 매우 의존적이다. 초기의 다중 객체 추적 기술은 이전 프레임에서 검출한 객체들의 위치 정보와 이후 프레임에서 검출한 객체 위치 정보 값을 비교하여 이들 간의 최소 차이들을 매칭하는 방식으로 진행되었다. 따라서 객체 검출이 제대로 이루어지지 않는 경우 이러한 객체 추적 방식은 객체들의 정확한 매칭이 어려워 추적의 정확성이 낮아질 수밖에 없다. 이러한 추적 문제를 해결하기 위해 객체 검출기의 성능을 높이는 연구 또한 여러 추적 문제 중에서 중요하게 다뤄지고 있다^[4]. 다른 측면으로는 동작 속도가 있는데, 최근 다중 객체 추적 시스템의 정확도는 객체 검출기의 발전으로 추적 정확도가 향상되어 상용 어플리케이션 구현이 가능한 정도이지만 많은 연산량으로 인해 실제 어플리케이션 구현에 어려움이 있다. 특히 포터블 디바이스 등 저성능 컴퓨팅 환경에서의 다중 객체 추적 시스템의 구현은 비 실시간성 동작 속도 문제로 다양한 응용 분야에서의 활용이 제한되고 있다^[4].

컴퓨터 비전 분야가 적용되는 감시, 활동 인식 또는 자율 주행 등과 같은 문제는 물체의 이동, 궤적에 대한 정보의 영향이 매우 크다. 다중 객체 추적에 대한 다양한 방법들이 이미 많이 제안되었지만, 일반적인 혼잡한 상황에서 여전히 검출 오류와 가려짐에 의한 추적 오류가 존재한다. 객체

추적 시스템의 구성이 대표적으로 객체 검출 - 추적의 형태임에 따라 최근 딥러닝 기반 검출, 추적기가 각각 적용되어 기존 대비 우수한 성능을 보이는 연구가 제안되었다^[9]. 본 논문에서는 이러한 딥러닝 모델이 적용된 다중 객체 추적 시스템에서 실시간 동작을 위한 방법을 제안하고 이에 대한 실험 결과를 공유한다.

기존 다중 객체 추적 시스템은 검출 모듈을 딥러닝 모델로 사용하고 이후 추적 모듈을 칼만 필터 등과 같은 궤적 예측 알고리즘으로 사용하여 구현했다. 최근에는 추적 모듈에도 Convolutional Neural Network(CNN) 기반의 딥러닝 모델을 사용하여 객체 추적 문제를 개선하는 연구가 진행되고 있다^[9]. 본 논문에서는 최근 딥러닝 기반 다중 객체 추적 알고리즘의 CNN 기반 객체 특징 추출 과정을 변형하여 CPU 환경에서 실시간성 동작 속도 구현을 위한 방법을 제안한다. 그림 1은 본 논문에서 제안하는 기법을 적용한 다중 객체 추적의 결과 영상이다.



그림 1. 다중 객체 추적 영상 결과
Fig. 1. Multi-object tracking video result

II. 관련 이론

1. 딥러닝 기반 객체 검출 방법

딥러닝은 신경망 네트워크로 많은 수의 계층을 만들어 학습하는 기계학습 분야이다. 신경망 네트워크는 오래전인 1950년대부터 인간의 뇌에서 영감을 얻어 시작되었지만 낮은 컴퓨팅 성능 문제와 XOR 연산 문제 등 탓으로 잠시 사라졌다가 최근 컴퓨팅 성능, 빅 데이터, Restricted Boltz-

mann Machine(RBM)으로 오버피팅 문제 해결 등을 통해 부활하였다. CNN의 특징 추출(Feature Extraction)은 컨볼루션 레이어와 풀링 레이어를 차례로 쌓은 형태로 되어있다. 컨볼루션 레이어는 필터와 같이 컨볼루션 연산을 통해 입력 영상을 변환하며, 풀링 레이어는 입력을 다운 샘플링하기 위해 적용되는 레이어로써 입력 값의 불필요한 정보를 없애고 압축하는 데 이용한다. 즉, 풀링 레이어는 이미지의 차원을 축소하는 역할을 한다. 컨볼루션 신경망인 CNN은 뇌의 시각 피질이 이미지를 처리하고 인식하는 원리를 이용한 신경망이다. 주로 영상 인식 분야에서 적용되며 여러 분야에서 탁월한 성능을 보여주고 있는 인공 신경망이다^[12]. CNN 기반 객체 검출 알고리즘 중 가장 잘 알려진 방식은 2014년에 발표된 Region-CNN(R-CNN)이다^[13]. R-CNN 알고리즘이 발표되기 이전에는 객체 검출을 위해 주로 Scale Invariant Feature Transform(SIFT)^[15], Histogram of Oriented Gradient(HOG)^[14], Optical Flow^[16], haar-like features^[17] 알고리즘 등이 자주 사용되었으며, 이는 영상에 존재하는 low-level feature에 기반을 두기 때문에 성능에 한계가 존재한다. 반면 R-CNN은 입력 영상에서 후보 영역을 만들고, 각 후보 영역에 대해 컨볼루션 신경망을 사용하여 해당 영상에 대해 분류를 하는 방법을 적용하여 기존 연구 대비 매우 높은 성능 향상을 보였다. R-CNN이 발표된 이후 R-CNN을 개선하기 위한 많은 연구가 진행되었는데, 대표적으로 SPP-Net^[18], Fast R-CNN^[19], Faster R-CNN^[20] 등이 있다. 하지만 R-CNN 계열의 객체 검출 네트워크는 높은 검출 성능을 보이지만 처리 속도가 매우 느린 단점을 가지고 있다. 후보 영역을 생성하고 각 후보에 대해 분류를 진행하고, 분류 결과 정보를 후처리를 통해 중

복 검출 결과를 제거하는 등 복잡한 파이프라인 구조로 되어있고, 개별 요소들이 분리되어 학습되므로 연산 양도 많고 최적화하기 어렵다. 이후 실시간 처리가 가능한 객체 검출 네트워크를 위해 복잡한 파이프라인 구조를 단일망에 해결하도록 설계된 Single Shot Multi-box Detector(SSD)^[21], YOLO^[10] 등의 검출 네트워크가 개발되었고 객체 검출 기반의 추적 알고리즘 또한 이러한 개발로 인해 점차 실시간 동작 구현의 가능성이 열렸다.

2. 칼만 필터 기반 객체 추적 시스템

칼만 필터는 잡음이 포함된 선형 역학계의 상태를 추적하는 재귀 필터로써 루돌프 칼만이 개발하였다. 칼만 필터는 컴퓨터 비전, 로봇 공학 등의 여러 분야에서 사용되고 있으며, 시간에 따라 진행한 측정을 기반으로 한다. 따라서 해당 순간에만 측정된 결과만 사용한 것보다는 좀 더 정확한 결과를 기대할 수 있다. 또한, 잡음까지 포함된 입력 데이터를 재귀적으로 처리하는 필터로써 현재 상태에 대한 최적의 통계적 예측을 진행할 수 있다. 칼만 필터는 선형 시스템을 기반으로 설계된 필터로써 정상상태 기준의 모델을 수행하였으므로, 선형 시스템이라고 가정할 수 있다. 칼만 필터의 알고리즘은 예측과 업데이트를 반복적으로 계산하여 시스템 출력 값의 잡음 영향을 최소화하고 출력 값을 예측한다. 예측은 현재 상태의 예측을 말하고, 업데이트는 현재 상태에서 관측된 측정까지 포함한 값을 통해서 더 정확한 예측을 할 수 있는 것을 말한다. 이러한 칼만 필터는 그림 2와 같이 동영상의 이전 프레임에서 객체 검출기를 통해 얻어진 객체의 위치 정보(경계 상자 좌표 x, y 와 상자

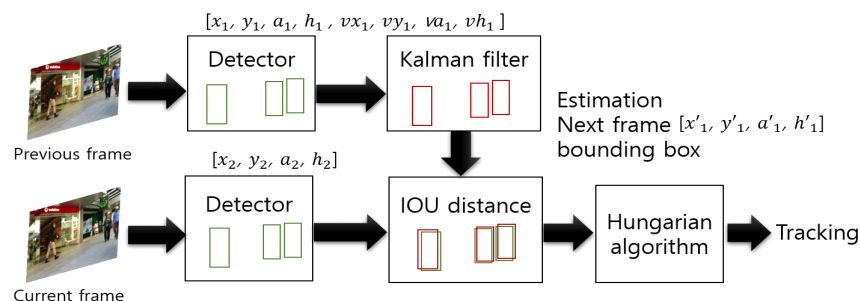


그림 2. 칼만 필터 기반 다중 객체 추적 시스템 개요

Fig. 2. Kalman filter-based Multi-Object Tracking System Overview

의 비율 a , 그리고 높이 h)를 통해 이후 프레임의 객체 위치 정보(x', y', a', h')를 예측한다. 확보한 예측 값은 이후 프레임에서 새롭게 검출한 객체와의 연결에 적용된다. 할당 비용 매트릭스는 각 검출과 기존 표적에서 예측된 모든 경계 상자 사이의 교차-분산(IOU)거리로 계산되고 헝가리안 알고리즘^[22]을 사용하여 최적으로 해결한다. 또한, 대상에 대한 검출이 IOU min보다 작은 할당을 거부하기 위해 최소 IOU가 부과된다. 딥러닝 이전에는 추적 알고리즘에서 주로 활용되었고 최근에도 딥러닝과 함께 많이 사용되고 있다.

III. 제안하는 기법

1. 딥러닝 기반 다중 객체 추적 시스템 개요

딥러닝 기반 다중 객체 추적 시스템은 기본적으로 CNN 기반의 네트워크를 사용하여 검출한 객체의 이미지 특징 정보를 추적 연결에 사용하는 방식이다^[9]. 그림 3을 통해 볼 수 있듯이 전후 프레임의 이미지 경계상자를 CNN 기반의 네트워크를 통해 특징정보를 추출하고 특징정보 비교기를 통해 추적 연결에 사용하는 방식이다. 기존의 칼만 필터 기반의 다중 객체 추적 시스템에 비해 정확도가 향상되었지만, 동작 속도가 느린 단점이 있다.

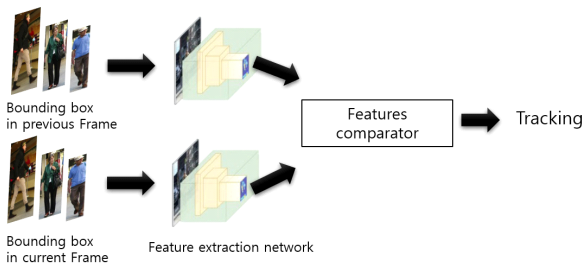


그림 3. 딥러닝 기반 다중 객체 추적 시스템 개요
Fig. 3. Deep Learning-based Multi-Object Tracking System Overview

최근 이러한 단점을 개선하고자 칼만 필터 방식과 딥러닝 기반 객체 추적 시스템을 혼용한 추적 알고리즘이 연구되었다(그림 4. 참조). 그림 4는 칼만 필터 방식과 딥러닝

기반 객체 추적 시스템을 혼용한 DEEP SORT^[9] 알고리즘의 개요이며 이는 기존 칼만필터를 통한 추적 방식^[11]에서 사용했던 IOU distance 알고리즘 대신 Mahalanobis distance 알고리즘을 적용하였고 이미지 특징 정보를 추출하는 CNN 기반 네트워크는 Deep cosine metric 알고리즘^[23]을 적용하여 구현하였다. 본 논문에서는 이 알고리즘 구조를 사용하여 실험하였고 동작 속도 측정 시 사용하는 객체 검출기는 사전 훈련된 YOLO-v3 모델^[10]을 사용하였다.

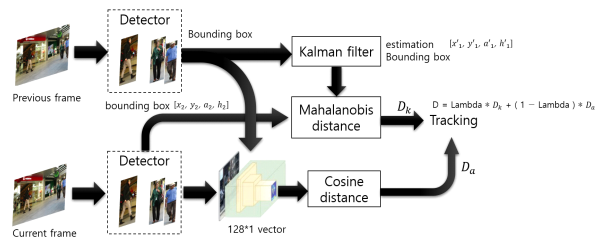


그림 4. 칼만 필터 방식과 딥러닝 기반 추적 방식을 혼용한 방법 개요
Fig. 4. Overview of tracking method using Kalman filter and deep learning

2. 프레임 스킵 기법

딥러닝 기반 객체 검출기는 연산량이 많으며 객체 검출-추적 시스템 특성 상 모든 입력 프레임에 대해 객체를 추적하는데, 저장된 영상들을 보면 프레임 레이트(1초 동안 보여주는 프레임의 수)가 10에서 30 혹은 그 이상으로 즉, 전후 프레임 간의 시간 차이가 0.6초 미만이다. 이는 일반적인 상황에서 프레임 사이의 정보 변화가 적고 모든 프레임에서 객체를 검출하는 것은 속도 측면에서 비효율적이라고 볼 수 있다. 그래서 본 저자는 객체 추적 시 모든 프레임에서의 객체 검출을 하지 않고 일부 프레임에서 객체 검출을 건너뛰는 방식으로 실행하면서 검출된 객체 정보를 추적 시스템에 적용하여 동작 속도를 향상시킬 수 있는지 실험하였다.

객체 검출 시 건너뛰는 검출 비율은 속도와 정확도 부분에서 그림 5와 같이 trade-off 관계를 나타낸다. 그림 5에서 왼쪽 세로축은 정확도(%)이고 오른쪽 세로축은 속도(frame per second; fps)를 뜻한다. 가로축은 기존 방법과 스킵 비율을 조절하여 비교한 것이며 5프레임당 1프레임, 4프레임당 1프레임, 3프레임당 1프레임 건너뛰기 방법을 비교하였다.

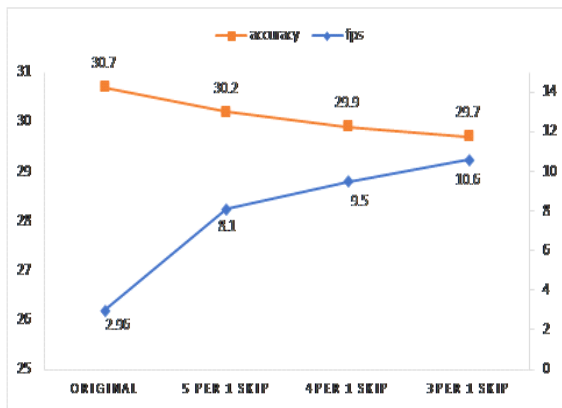


그림 5. 검출 프레임 비율에 따른 정확도와 속도 상관관계

Fig. 5. Accuracy and Speed Correlation with Detection Frame Rate

1 프레임 당 1 프레임, 2 프레임 당 1 프레임 건너뛰기 방법은 추적 시스템의 칼만 필터 모듈에서 정확도가 매우 낮아져, 다중 객체 추적 정확도 측정 실험에서 제외하였다.

3. 여러 이전 프레임 정보를 추가 업데이트하는 방법

그림 6은 본 논문에서 두 번째로 제안하는 여러 이전 프레임 정보를 추가 업데이트하여 추적에 연결하는 방법의 개요 이미지이다. 기존 제안된 딥러닝 기반 다중 객체 추적 시스템^[9]에서는 이전 프레임에서 검출한 객체의 이미지 특징 정보와 이후 프레임에서 검출한 객체의 이미지 특징 정보를 비교하여 차이 값이 가장 작은 객체끼리 맵핑하여

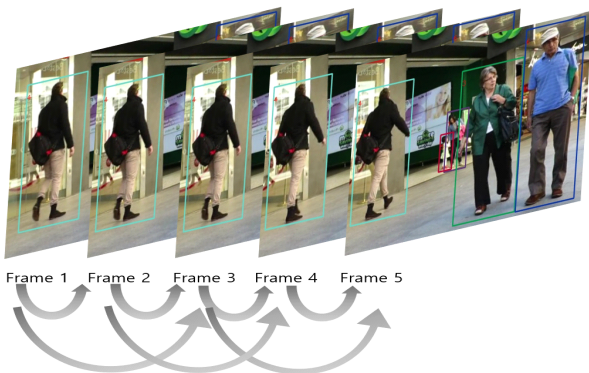


그림 6. 여러 이전 프레임 정보를 추가 업데이트하는 방법

Fig. 6. Method to update previous to previous frame

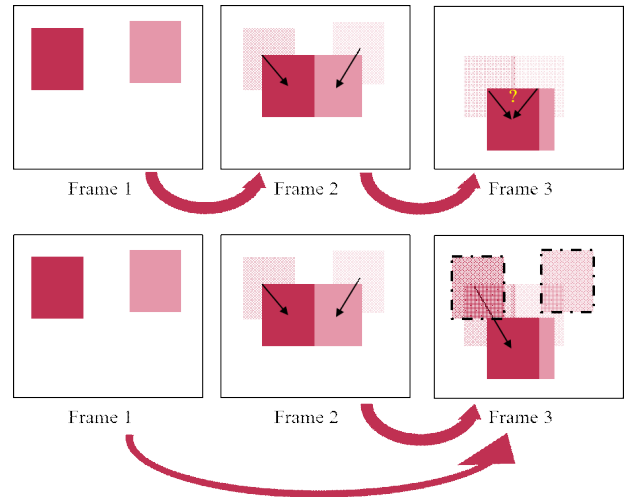


그림 7. 비슷한 이미지 특징 정보를 가진 두 객체가 겹칠 때 기존 방법과 제안하는 방법 비교

Fig. 7. Comparison between existing and proposed methods when two objects have similar image features

추적을 완성한다. 하지만 기존 방식은 그림 7의 위쪽 부분 Frame 3에서 비슷한 이미지 특징 정보를 가진 두 객체가 겹칠 때 추적 연결 시 어려움을 갖는다. 그림 7의 아랫부분 처럼 하나 더 이전 프레임의 정보를 이용한다면 거리 차이가 더 분명하고 객체의 뒤 배경 차이가 더 크기 때문에 이미지 특징 정보 값의 차이도 더 큰 차이를 갖기에 더 정확한 추적 연결이 가능하다. 또한, 이전 프레임에서 검출했던 정보를 저장해놨다가 재이용하는 것이기 때문에 추가적인 검출 연산량이 필요하지는 않다. 그러한 이유로 추가 실험 결과에서 동작 속도가 크게 줄어들지 않은 결과를 볼 수 있다 (표 2. 참조). 추적 연결에 대한 수학적 표현은 수식 (1)과 같다. $O_t = (o_t^1, o_t^2, \dots, o_t^n)$, O_t 는 t 번째 프레임에서 검출한 객체들의 집합이며, I_t 는 t 번째 프레임에서의 객체들의 id 값이다. 현재 프레임과 이전 프레임의 객체들의 feature 값의 유사성과 현재 프레임과 하나 더 이전 프레임의 객체들의 feature 값의 유사성을 비교하여 유사성이 더 높은 프레임에서의 객체와 같은 id를 현재 프레임의 객체에 id를 부여하여 연결한다. C 는 deep cosine metric 알고리즘이며^[23], 객체들의 feature 유사성을 연산한다.

$$I_t = \max(C(O_t, O_{t-2}), C(O_t, O_{t-1})) \quad (1)$$

IV. 실험 결과

1. 실험 환경 및 절차

본 논문에서는 MOT16^[8] 객체 추적 영상 데이터셋을 사용해 실험하였다. 실험은 Intel(R) Xeon(R) CPU E5-2687W v4 @ 3.00GHz 12 core CPU 환경에서 진행하였다. 기존 방법과 제안하는 방법의 성능을 평가하기 위해 표준 평가 지표인 CLEAR MOT^[6]를 포함한 7개의 지표를 사용하여 종합적으로 평가하였다.

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + IDsw_t)}{\sum_t GT_t} \quad (2)$$

$$MOTAL = 1 - \frac{\sum_t (FP_t + FN_t + \log_{10}(IDsw_t + 1))}{\sum_t GT_t} \quad (3)$$

위의 방정식에서 ‘GT’는 ground truth를 의미하며 방정식에 사용된 다른 기호는 표 1에서 언급한다.

표 1. 벤치마크에서 사용하는 함수

Table 1. Metrics used for benchmarking

Metric	Better	Perfect	Description
MOTA	higher	100%	Overall Tracking Accuracy.
MOTAL	higher	100%	Log Tracking Accuracy.
MOTP	higher	100%	Percentage alignment of predicted bounding box and ground truth.
FP	lower	0	Number of false positives.
FN	lower	0	Number of false negatives.
IDsw	lower	0	Identity switches, see [80] for details.
FPS	higher	Inf.	Processing speed in frames per second.

2. 제안하는 방법 실험 결과

표 2는 기존 알고리즘과 제안하는 기법 두 가지를 적용하

표 2. 기존 방법과 2가지 제안 방법 적용 실험 결과

Table 2. Experimental results applying the original method and the two proposed method

METHOD	VIDEO	MOTA ↑	MOTP ↑	MOTAL ↑	FP ↓	FN ↓	IDs ↓	FPS ↑
original	MOT16-02	<u>16.5</u>	76.5	<u>16.7</u>	<u>545</u>	<u>14307</u>	40	3.3
	MOT16-04	34.9	79.4	35.2	<u>3051</u>	<u>27751</u>	157	3.08
	MOT16-05	<u>29.5</u>	77.1	<u>30</u>	<u>332</u>	<u>4440</u>	32	5.3
	MOT16-09	47.2	74.8	48.1	<u>481</u>	<u>2244</u>	<u>53</u>	4.5
	MOT16-10	<u>32</u>	75.8	32.3	<u>634</u>	<u>7701</u>	<u>36</u>	3.2
	MOT16-11	46.1	79.5	46.4	<u>654</u>	<u>4266</u>	<u>23</u>	4.6
	MOT16-13	15	74.4	15.1	<u>170</u>	<u>9546</u>	<u>22</u>	3.3
	AVERAGE	30.7	78	31.1	<u>5867</u>	<u>70255</u>	<u>363</u>	3.9
frame skip	MOT16-02	15.9	76.3	16.1	437	14520	41	12.6
	MOT16-04	<u>34.2</u>	79.4	<u>34.5</u>	2509	28653	132	11.9
	MOT16-05	27.6	76.5	27.9	220	4693	20	49.8
	MOT16-09	46.4	74.8	47.2	364	2409	47	12.3
	MOT16-10	30	<u>75.7</u>	30.3	486	8099	38	31.8
	MOT16-11	45.8	79	<u>46.1</u>	516	4428	25	12.9
	MOT16-13	12.7	73.9	12.8	133	9845	18	12.6
	AVERAGE	29.7	<u>77.9</u>	30	4664	72647	321	20.6
frame skip + previous to previous frame	MOT16-02	16.6	<u>76.4</u>	16.8	605	14237	39	<u>11.8</u>
	MOT16-04	34.1	<u>79.3</u>	<u>34.5</u>	3618	27524	175	<u>11.1</u>
	MOT16-05	29.8	<u>76.9</u>	30.2	395	4362	<u>30</u>	<u>49.6</u>
	MOT16-09	<u>46.5</u>	<u>74.7</u>	<u>37.4</u>	543	2218	54	<u>11.2</u>
	MOT16-10	32.1	75.7	32.3	683	7651	34	<u>12.2</u>
	MOT16-11	<u>45.5</u>	79.5	45.7	742	4238	22	<u>12.1</u>
	MOT16-13	<u>14.9</u>	<u>74.3</u>	15.1	234	9488	24	<u>12.2</u>
	AVERAGE	<u>30.3</u>	<u>77.9</u>	<u>30.7</u>	6820	69718	378	<u>17.2</u>

여 실험한 결과표이다. 기존 알고리즘은 CPU 환경에서 평균 약 4 fps 동작 속도를 보이며 실시간 동작과는 다소 거리가 멀다. 첫 번째 제안 방법인 객체 검출 프레임 비율을 조절하는 방법은 3개 프레임 간격마다 1개 프레임의 객체 검출을 건너뛰는 비율을 적용하여 객체 검출 연산량을 줄였고 평균 20 fps인 실시간 동작 속도를 얻을 수 있었지만, 정확도 측면에서는 0.3 퍼센트 정도, 다소 낮아진 결과를 얻었다. 실험 영상 중, 객체 간 가려짐이 많이 발생하는 MOT16-13 영상에서는 가장 낮은 정확도를 보였다. 이러한 결과는 추적에 연결하는 데이터양의 감소에 의한 것으로 보이며, 추가적으로 객체 이동 속도가 빠르거나 다른 객체의 가려짐이 여러 프레임 동안 발생한 경우로 분석된다. 두 번째 제안 방법인 여러 개의 이전의 프레임 정보를 추가 업데이트하는 방법을 이전 실험에 추가 적용하여 추가 실험 결과를 확인했다. 하나 더 이전의 프레임 정보를 추가 업데이트 시 객체 검출기를 통해 재검출하는 것이 아니라 이전 프레임에서 검출했던 정보를 저장해놔다가 재이용하는 것이기 때문에 추가적인 검출 연산량이 필요하지는 않다. 그러한 이유로 추가 실험 결과에서 동작 속도가 크게 줄어들지 않은 결과를 볼 수 있다. 또한, 본 실험을 통해 trade-off 관계인 정확도와 동작 속도의 반 비례성을 최소화 시켰고 테스트 영상에 따라 객체 검출 프레임을 건너뛰더라도 더 높은 종합 정확도(MOTA)를 얻는 결과를 확인할 수 있다. (표 2.에서 진한 숫자는 가장 높은 수치를 뜻하며 밑줄 친 숫자는 두 번째로 높은 수치를 의미한다(표 2의 약어들은 표 1 참조).

V. 결 론

본 논문에서는 기존 객체 검출-추적 기반의 연쇄된 형태의 알고리즘에서 객체 검출 정도를 조절하여 CPU 환경에서의 실시간 동작을 위한 방법을 제안한다. 매 프레임마다 객체 검출을 수행하는 대신 3개 프레임 간격마다 1개 프레임의 객체 검출을 건너뛰는, 객체 검출 프레임 빈도수를 조절하는 방법으로 전체 추적기의 동작 속도를 실시간 처리가 가능한 수준까지 개선하였다. 또한, 하나 더 이전의 프레임 정보를 추적 알고리즘에 업데이트하여 추가 비교함으로

써 추적 정확성을 향상시킨 방법을 제안한다.

참 고 문 헌 (References)

- [1] Betke, Margrit, Esin Haritaoglu, and Larry S. Davis. "Real-time multiple vehicle detection and tracking from a moving vehicle." *Machine vision and applications* 12.2 (2000): 69-83.
- [2] Hu, Weiming, et al. "A survey on visual surveillance of object motion and behaviors." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 34.3 (2004): 334-352.
- [3] Lu, Wei-Lwun, et al. "Learning to track and identify players from broadcast sports videos." *IEEE transactions on pattern analysis and machine intelligence* 35.7 (2013): 1704-1716.
- [4] Murray, Samuel. "Real-time multiple object tracking-A study on the importance of speed." *arXiv preprint arXiv:1709.03572* (2017).
- [5] Luo, Wenhan, et al. "Multiple object tracking: A literature review." *arXiv preprint arXiv:1409.7618* (2014).
- [6] Bernardin, Keni, and Rainer Stiefelhagen. "Evaluating multiple object tracking performance: the CLEAR MOT metrics." *EURASIP Journal on Image and Video Processing* 2008 (2008): 1-10.
- [7] Ristani, Ergys, et al. "Performance measures and a data set for multi-target, multi-camera tracking." *European Conference on Computer Vision*. Springer, Cham, 2016.
- [8] Milan, Anton, et al. "MOT16: A benchmark for multi-object tracking." *arXiv preprint arXiv:1603.00831* (2016).
- [9] Wojke, Nicolai, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric." *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017. doi: 10.1109/ICIP.2017.8296962
- [10] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [11] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016, pp. 3464-3468. doi: 10.1109/ICIP.2016.7533003
- [12] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324. doi: 10.1109/5.726791
- [13] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
- [14] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. IEEE, 2005.
- [15] Lowe, David G. "Object recognition from local scale-invariant features." *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. IEEE, 1999.
- [16] Horn, Berthold KP, and Brian G. Schunck. "Determining optical flow." *Techniques and Applications of Image Understanding*. Vol. 281.

- International Society for Optics and Photonics, 1981.
- [17] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Vol. 1. IEEE, 2001.
- [18] He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37.9 (2015): 1904-1916.
- [19] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
- [20] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.
- [21] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.
- [22] Kuhn, Harold W. "The Hungarian method for the assignment problem." Naval research logistics quarterly 2.1 2 (1955): 83-97.
- [23] Wojke, Nicolai, and Alex Bewley. "Deep cosine metric learning for person re-identification." 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, 2018.

저 자 소 개



김 경 훈

- 2019년 : 연세대학교 컴퓨터공학과 학사
- 2019년 : 서강대학교 전자공학과 석사과정
- ORCID : <https://orcid.org/0000-0002-3196-3470>
- 주관심분야 : 컴퓨터 비전, 딥러닝, 다중 객체 추적



허 준 호

- 2018년 : 광운대학교 전자융합공학과 학사
- 2018년 ~ 2020년 : 서강대학교 전자공학과 석사
- ORCID : <https://orcid.org/0000-0002-7575-0354>
- 주관심분야 : 딥러닝, 컴퓨터 비전 시스템 설계



강 석 주

- 2006년 : 서강대학교 전자공학과 학사
- 2011년 : 포항공과대학교 전자전기공학과 공학박사
- 2011년 ~ 2012년 : LG Display 선임 연구원
- 2015년 ~ 현재 : 서강대학교 전자공학과 부교수
- ORCID : <https://orcid.org/0000-0002-4809-956X>
- 주관심분야 : 멀티미디어 영상신호처리, 컴퓨터 비전, 딥러닝 시스템 설계