Journal of KIIT. Vol. 17, No. 11, pp. 1-8, Nov. 30, 2019. pISSN 1598-8619, eISSN 2093-7571 1
http://dx.doi.org/10.14801/jkiit.2019.17.11.1

실시간 응용을 위한 혼합형 다중 객체 추적 시스템의 구현

우윤희*, 최권택**, 이정근***

Hybrid Multiple Object Tracking System Design for Real-Time Application

Yun-Hee Woo*, Kwon-Taeg Choi**, and Jeong-Gun Lee***

본 연구는 중소벤처기업부와 한국산업기술진흥원의 "지역특화산업육성(R&D, P0004928)" 사업의 지원을 받아 수행된 연구결과임

요 약

객체 추적(Object Tracking)은 최근 몇 년간 딥 러닝의 발전에 영향을 받아 급격하게 발전했다. 딥 러닝을 통해 학습된 객체 검출 시스템과 결합된 객체 추적 시스템들은 뛰어난 성능을 보여주지만 고성능의 연산 시스템을 필요로 한다. Simple Online and Realtime Tracking(SORT)는 다중 객체 추적 알고리즘으로 별도의 객체 검출 시스템과 결합하여 이용된다. SORT는 적은 수의 객체를 추적하는 상황에서도 비교적 많은 연산량을 필요로 한다. 본 논문에서는 SORT를 유클리드 거리를 이용한 객체 추적 방법과 결합한 혼합형 추적 시스템을 제안하고 제안된 방법으로 상황에 따라 적응적으로 추적 방법을 바꾸는 객체 추적 알고리즘을 구현하였으며 기존의 SORT와 비교하여 효과적으로 연산량을 줄였음을 보인다.

Abstract

In recent years, object tracking techniques have grown dramatically with the advances of deep learning. Object tracking systems combined with deep learning show good tracking performance but they require high computational cost. Simple Online and Realtime Tracking(SORT) is a multiple object tracking algorithm that is used with an object detection system. SORT requires a large amount of operations even when just few objects in distance need to be tracked. In this paper, we propose a system that reduces the amount of computation by combining SORT with the light-weight object tracking method using Euclidean distance. We implement the hybrid object tracking algorithm that changes the tracking method flexibly according to the situation. Finally we show that the amount of computation for the proposed algorithm is effectively reduced compared to the SORT.

Keywords
computer vision, multiple object tracking, data association, low computational system

* 한림대학교 컴퓨터공학과 박사과정
- ORCID: http://orcid.org/0000-0003-2240-122X
** 강남대학교 소프트웨어응용학부 교수
- ORCID: http://orcid.org/0000-0001-5331-321X
*** 한림대학교 소프트웨어융합 교수(교신저자)
- ORCID: http://orcid.org/0000-0001-6218-4560

· Received: Oct. 22, 2019, Revised: Nov. 04, 2019, Accepted: Nov. 07, 2019
· Corresponding Author: Jeong-Gun Lee
School of Software, 1211 Sungho-Bild., Hallym University,
39 Hallymdaehak-gil, Chuncheon,Gangwon-do, 200-702 Korea
Tel.: +82-33-248-2312, Email: jeonggun.lee@hallym.ac.kr

Ⅰ. 서 론

인류 사회가 복잡해지면서 생활 안전 등을 위해 최근 다양한 곳에서 CCTV를 쉽게 찾아볼 수 있다. CCTV가 설치되는 주된 목적은 도난 방지 등의 방범을 위해서이지만 그 외에도 여러 목적을 위해 응용될 수도 있다. 예를 든다면 CCTV는 항상 행인 또는 고객의 행동을 기록하고 있으므로 그들의 행동 패턴에 대한 자료를 획득할 수 있다. 가게 또는 행사장과 같이 사람들의 수요가 무엇인지 분석할 필요가 있는 장소에서 사람들이 많은 관심을 갖는 부분은 어떤 것인지, 사람들의 이동 흐름은 어떠한지 등의 중요한 정보를 CCTV 영상 분석을 통해서 얻을 수 있다. 이러한 정보를 얻기 위해서는 단순히 영상을 저장하는 것뿐만 아니라 영상을 효율적으로 분석하는 과정이 필요하다.

영상을 분석하는 것은 컴퓨터 비전 분야의 일이다. 그 중에서도 앞서 말한 예시와 같은 경우에는 객체 검출(Objecct detection)및 객체 추적(Object tracking)이 필요하다. 사람들이 무엇에 관심을 갖고 어디로 이동하는지에 대한 정보를 얻기 위해서는 객체 추적이 수행되어야 하며 객체 추적을 위해서는 객체 검출이 선행되어야 한다. 딥 러닝의 발전에 힘입어 객체 검출과 객체 추적은 최근 컴퓨터 비전에서 가장 많은 관심을 끌고 있는 분야이다. 최근 몇 년 간 기계 학습은 놀라운 속도로 발전했으며 컴퓨터 비전 분야는 그 혜택을 직접적으로 받았다. 초기 MNIST, CIFAR-10과 같은 비교적 간단한 분류 문제가 대상이었다. 하지만 점차 객체 검출과 같은 복잡한 문제에도 적용되기 시작하였으며 그 처리 속도 또한 매우 빨라져 실시간 객체 검출 시스템[1, 2]까지 등장하게 되었다.

객체 추적에서 객체를 정확히 검출하는 것은 중요하지만 그것과 마찬가지로 중요한 것이 데이터 연계(Data association) 문제이다. 객체 검출을 통해 얻은 결과를 이전 프레임과 비교하여 동일한 객체끼리 연결함으로써 객체 추적이 수행되도록 해야 한다. SORT(Simple Online and Realtime Tracking)[3]는 실시간 객체 추적을 가능하게 하는 다중 객체 추적 시스템이다. SORT는 칼만 필터[4]와 헝가리안 알고리즘[5]을 통해 다음 프레임에서 객체의 위치를 예측하고 예측 결과를 검출된 객체와 연계하여 객체 추적을 수행한다. 그 성능은 뛰어나지만 칼만 필터를 이용한 예측 방법은 많은 연산을 필요로 한다.

이에 반해 “유클리드 거리”(Euclidean distance)를 이용한 객체 추적 방법은 매우 적은 연산량을 요구하는 경력 추적 방법이다. 이 방법은 단순히 이전 프레임과 현재 프레임의 객체들 사이의 거리를 연산하여 가장 가까운 객체들끼리 매칭시키는 방법인데 연산량이 매우 적다는 장점이 있다. 하지만 “객체 스위칭(Identity switch)”이라는 단점 역시 존재한다. 두 객체가 서로 교차할 때 이전 프레임과 비교하여 가장 가까운 객체가 자기 자신이 아닌 엇갈려 지나간 상대일 경우 객체 스위칭이 발생하게 된다. 반면 칼만 필터를 통해 예측을 했을 경우 단순히 객체 사이의 거리를 계산한 것이 아닌 다음 프레임에서 객체의 위치를 예측한 것이므로 객체 스위칭이 잘 발생하지 않는다. 본 논문에서는 이와 같은 두 개 상이한 특성을 갖는 방법의 장단점에 착안하여 두 방법을 결합한 혼합형 시스템(Hybrid system)을 제안한다. 객체 스위칭이 발생할 우려가 없는 경우, 즉 객체들이 서로 일정 거리 이상 떨어져있을 때에는 유클리드 거리를 이용한 방법을 이용한다. 객체들이 일정 거리 이내로 가까워져 객체 스위칭이 발생할 위험이 있는 경우에는 SORT를 적용한다. 이를 통해 기존 SORT에서 매 프레임 칼만 필터를 호출함에 따라 발생했던 불필요한 연산을 줄인다.

본 논문의 2장에서는 관련 연구로 객체 검출 시스템과 SORT에 대해 기술하였으며 3장에서는 제안된 저연산 혼합형 객체 추적 시스템에 대해 설명한다. 4장은 제안된 시스템을 기존 SORT의 결과와 비교 및 분석하며 마지막으로 5장에서 결론 및 한계점에 대해 기술한다.

Ⅱ. 관련 연구

2.1 객체 검출 시스템

객체 검출 시스템은 본 논문에서 제안된 시스템을 구현하는 데에 있어 필수 요소이다. 객체 검출 시스템은 크게 2-단계(Two-stage) 기법과 1-단계(One-stage) 기법의 두 가지 분류로 나누어진다.

2-단계 기법은 1-단계 기법이 주목을 받기 전 주목을 받았고 많은 연구가 진행되었던 기법이다. 2-단계라는 이름에서 알 수 있듯이 두 단계에 걸쳐 객체 검출을 수행한다. 첫 번째는 주어진 영상에서 객체가 있을 것으로 예상되는 영역을 추출하는 영역 제안(Region proposal) 단계이며 두 번째는 이전 단계에서 제안된 영역에 있는 객체가 어떤 종류인지 판단하는 분류 단계이다. 이 두 가지 단계를 거쳐야 하나의 영상에 대한 객체 검출이 완료되었기에 2-단계 기법들은 대부분 느린 검출 속도를 갖는다. 하지만 하나의 객체가 있을 것이라 추정되는 영역 제안과 객체 분류가 따로 진행되기 때문에 높은 정확도를 갖는다. 2-단계 기법들은 검출 속도를 높이는 것에 중점을 두기도 하는데 대표적인 예로 R-CNN[6], Fast R-CNN[7], Faster R-CNN[2]이 존재한다. 가장 최근에 제안된 Faster R-CNN의 경우 실시간 객체 검출에도 적용할 수 있을 정도로 검출 속도가 증가했지만 여전히 1-단계 기법의 객체 검출 시스템에 비하면 느린 속도를 갖는다.

실시간 객체 검출 분야에서 2-단계 기법이 주된 주제일 때 나타난 “You Only Look Once”(YOLO) [1]라는 1-단계 기법이 적용되었고, 이후 많은 주목을 이끌었다. 1-단계 기법은 영역 제안과 객체 분류가 동시에 이루어지기 때문에 이러한 이름을 갖게 되었다. 2-단계 기법에 비해 매우 빨라 실시간 객체 검출을 가능하게 하지만 처리가 동시가 이루어지는 만큼 그 정확도는 2-단계 기법의 시스템들에 비해 낮은 편이다.

본 논문에서는 실시간 객체 추적을 위한 객체 검출 시스템으로 YOLOv3를 이용한다. 객체 추적 시스템에서는 객체 검출이 선행된 후 그 결과를 기반으로 데이터 연계을 수행해야 한다. 때문에 실시간으로 이러한 과정이 진행되기 위해서는 객체 검출 또한 매우 빠른 속도로 처리되어야 한다. YOLO는 이러한 과정에 적합하기 때문에 선택되었다.

2.2 SORT

SORT는 온라인 객체 추적 시스템이다. 객체 추적 시스템은 두 가지 분류로 나눌 수 있는데 하나는 배치방법이고 또 다른 하나는 온라인 방법이다. 배치 방법으로 동작하는 객체 추적 시스템은 미래의 정보를 알아야 한다. 즉 실시간 시스템에서는 사용할 수 없다. 반면 온라인 방법은 현재와 이전 프레임의 정보만으로 객체 추적을 수행하기 때문에 실시간 객체 추적 시스템 구현이 가능하다.

SORT는 칼만 필터를 이용하여 다음 프레임에서의 객체의 위치를 예측한다. 칼만 필터는 과거의 측정값을 바탕으로 현재 상태를 예측하는 알고리즘으로 예측과 업데이트를 두 단계를 번갈아 수행하며 동작한다. 예측 단계에서는 이전의 측정값을 바탕으로 현재 상태를 예측하며 업데이트 단계에서는 현재 측정값을 입력으로 받아들여 변수를 업데이트한다. 우선 칼만 필터의 예측 단계는 아래의 수식으로 표현된다.

$$\widehat{x_k^-} = A\hat{x}_{k-1} + Bu_k \tag{1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2}$$

위 식에서 k는 시간, x_k는 k시간에서의 상태, A와 B는 상태 전이 행렬이다. u는 입력을 의미하지만 현재 객체 추적 시스템에서는 고려하지 않는다. P는 오차 공분산 행렬, Q는 시스템 잡음이다. 다음으로 업데이트 단계의 수식은 아래와 같다.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{3}$$

$$\widehat{x_k} = \widehat{x_k^-} + K_k(z_k - H\widehat{x_k^-}) \tag{4}$$

$$P_k = (I - K_k H)P_k^- \tag{5}$$

K는 칼만 이득, H는 측정 행렬, R은 측정 잡음 공분산 행렬, z는 측정값을 의미한다. 이 때 상태 x는

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{6}$$

위와 같으며 이 때 u는 객체의 중심의 x축 좌표값, v는 y축 좌표 값, s는 바운딩 박스(Bounding

box)의 크기, r은 바운딩 박스의 가로, 세로 비율이다.

칼만 필터는 두 단계가 번갈아가며 예측과 업데이트를 반복한다. 이렇게 계산된 예측 결과는 실제 검출 결과와 비교된다. 가장 높은 IoU(Intersection of Union)를 갖는 객체들은 같은 객체로 취급되며 검출 결과와 매칭되지 못한 객체는 일정시간이 지난 후에도 업데이트 되지 못하면 사라진 객체로 판단하고 삭제된다. 검출 결과 새로운 객체가 등장하면 새 객체를 위한 추적기를 생성한다. 매 프레임 이러한 과정을 거치기 때문에 SORT는 정확도가 높지만 연산량이 많다.

III. 저연산 실시간 다중 객체 추적 시스템

객체 추적 시스템을 구현하는 방법 중 가장 쉬운 방법 중 하나는 유클리드 거리를 이용하는 방법이다. 단순히 프레임간 검출된 객체들 사이의 거리를 계산하여 가장 가까운 객체가 무엇인지 파악한다. 유클리드 거리를 구하는 공식은 아래와 같다.

$$d = \sqrt{(x_{curr} - x_{prev})^2 + (y_{curr} - y_{prev})^2} \tag{7}$$

여기서 x_{curr}과 y_{curr}은 현재 프레임에서 객체의 중심의 x, y 축 좌표이며 x_{prev}와 y_{prev}는 이전 프레임에서의 객체의 중심의 x, y 축 좌표이다.

표 1은 객체 추적 방법에 따른 이론적 연산량 분석 식을 보여준다. 연산량 분석 식에서 m은 이번 프레임에서 검출된 객체 수, n은 이전 프레임에서 검출된 객체 수를 의미한다.

표 1. 객체 추적 방법에 따른 연산 복잡도
Table 1. Computational complexity according to object tracking method

Tracking method / Operation	SORT	Euclidean distance
+	1711×n+1×m×n	1×m×n
−	69×n+7×m×n	2×m×n
*	2027×n+3×m×n	2×m×n
/	1×m×n	0
$\sqrt{\ }$	0	1×m×n
compare	6×m×n+(m−1)1×m×n	1×m×n

그림 1과 그림 2는 표 1에 표기된 연산량을 그래프로 나타낸 것이다. 그림 1은 m과 n이 같을 경우, m 또는 n이 증가할 때의 연산량의 변화를 나타내었고 그림 2는 m/n의 비율에 따른 연산량의 변화를 나타내었다.

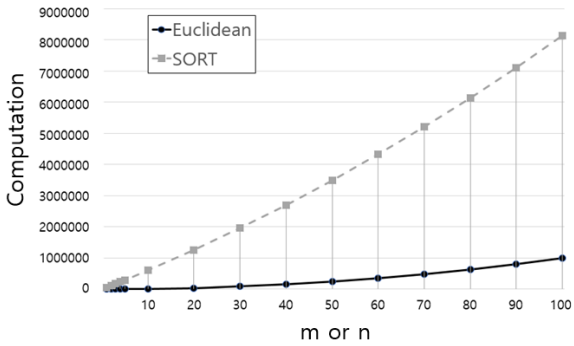


그림 1. m, n의 증가에 따른 연산량의 변화
Fig. 1. Amount of computation as m, n increase

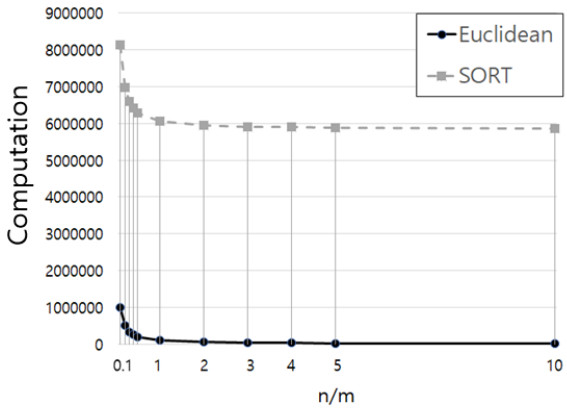


그림 2. m, n의 비율 변화에 따른 연산량의 변화
Fig. 2. Amount of computation as ratio of m, n changes

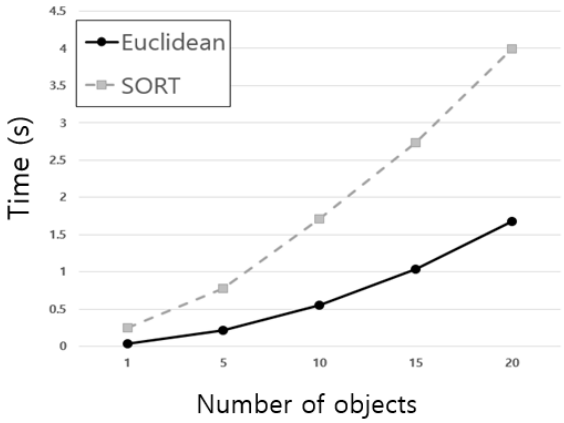


그림 3. 객체 수에 따른 실제 연산 시간의 차이
Fig. 3. Real computation times as the number of objects increases

또한, 그림 3은 객체 수에 따른 두 방법의 실제 연산 시간 차이를 보여준다. 세 가지 그래프에서 알 수 있듯이 두 방법의 연산량의 차이는 극명하다.

칼만 필터는 객체가 많아 복잡한 상황에서도 좋은 결과를 보여주는 알고리즘이지만 객체 추적에서 항상 필요한 것은 아니다. 사람들의 밀집도가 적은 상황에서는 유클리드 거리를 이용한 추적법도 충분한 성능을 보여준다. 사람의 수가 많은 상황이 아니라면 연산량을 보았을 때 오히려 유클리드 거리를 이용한 방법이 더 효율적이다. 하지만 사람의 수가 적다고 해도 교차는 흔히 발생하며 갑자기 다수의 사람이 등장할 수도 있다. 이럴 경우에는 칼만 필터를 이용하는 것이 정확도 측면에서 더 좋은 결과를 보여준다. 따라서 본 논문에서는 상황에 따라 유동적으로 칼만 필터와 유클리드 거리를 이용한 방법을 교체해가며 사용하는 객체 추적 시스템을 제안한다. 제안하는 시스템의 알고리즘은 그림 4로 설명된다.

Algorithm 1 Low computation multiple object tracking

For each frame,
1: Get detection result of the current frame from YOLO
2: Calculate Euclidean distance *d* between objects from the current frame and the previous frame
3: **if** more than one object has *d* less than half the height of the object **then**
4: Set the Euclidean delay to zero
5: **if** previous tracking method is Euclidean **then**
6: Increase the Kalman delay by one
7: **if** the Kalman delay is smaller than one **then**
8: **go to** 18
9: Use the Kalman filter to get the predicted location of objects
10: Match the predicted location and the detection result with the Hungarian algorithm
11: Update the Kalman filter
12: **else**
13: Set the Kalman delay to zero
14: **if** previous tracking method is SORT **then**
15: Increase the Euclidean delay by one
16: **if** the Euclidean delay is smaller than three **then**
17: **go to** 9
18: Compare the detection result with the previous frame to match the objects with the shortest *d*
19: **end if**
20: Save the tracking result of the current frame

그림 4. 저연산 다중 객체 추적 시스템
Fig. 4. Low computation multiple object tracking

알고리즘의 내용은 다음과 같다. 우선 매 프레임 YOLO를 이용하여 프레임 내의 객체들을 검출한다. 그 후 이전 프레임의 객체들과 현재 프레임의 검출된 객체들 사이의 유클리드 거리를 계산한다. 객체들이 서로 가까운 지 판단하는 기준은 각 객체의 바운딩 박스의 높이가 된다. 이전 프레임의 객체를 기준으로 해당 객체의 바운딩 박스의 높이의 절반보다 가까운 거리에 두 개 이상의 현재 프레임의 객체가 탐지된다면 객체 스위칭이 발생할 수 있으므로 이 때는 SORT로 추적 방법을 교체해준다. 기준을 바운딩 박스의 높이 절반으로 정한 이유는 일반적으로 사람의 보폭은 자신의 키의 절반에 못 다다르는 정도이기 때문이다. 이 때 객체 탐지기에 의해 사람이 아닌 객체가 사람으로 탐지되는 경우도 있으므로 추적 방법 교체로 발생하는 성능 저하를 줄이기 위해 한 프레임동안 추적 방법을 교체하지 않는다. 그 후에도 일정 거리 안에 다수의 객체가 탐지된다면 유클리드 거리를 이용한 추적 방법으로 교체한다. 만약 두 사람이 스쳐 지나간 뒤 멀리 떨어졌다면 추적 방법은 유클리드 거리를 이용한 방법으로 교체된다. 이 때 또한 객체 탐지기에 의해 탐지되어야 할 객체가 탐지 되지 않는 경우가 있으므로 일정 프레임동안 기존 추적 방법을 유지한다. 추적 방법을 유클리드 거리를 이용한 방법에서 SORT로 교체할 시 기존 SORT의 정보는 모두 삭제된다. SORT는 매 프레임 정보를 업데이트 시켜 정확도를 향상시키기 때문에 정보 삭제는 치명적이다. 따라서 이 경우는 세 프레임동안 SORT를 유지한다. 그 후에도 SORT가 필요하지 않다고 판단되면 유클리드 거리를 이용한 방법으로 교체된다.

두 방법은 서로 다른 추적기를 사용하지만 이전 프레임의 정보를 필요로 하는 것은 마찬가지이다. 따라서 매 프레임 추적이 완료된 후에는 추적 방법 교체 시 서로 결과를 공유하기 위해 추적 결과를 별로도 저장하는 작업이 필요하다. 이 때 10 프레임 동안 업데이트되지 않은 객체는 영상에서 사라진 것으로 간주하고 목록에서 삭제하도록 하였다.

Ⅳ. 실험 및 결과 분석

본 논문에서 제안하는 시스템이 큰 효과를 보는

상황은 사람들이 유동량 변화가 큰 경우이다. 사람들이 밀집된 경우 정확성을 위해서 항상 칼만 필터를 이용함으로써 밀집된 상황이후 발생할 수 있는 한적한 상황에서 발생하는 불필요한 연산들이 생기는 경우 많다. 본 논문에서 제안한 시스템을 평가실험하기 위해서 연산량이 적은 경우와 많은 경우가 적절하게 혼합된 상황에 대한 영상 데이터, 즉 사람들이 일정한 거리를 유지하고 있으면서 때로는 엇갈려 지나가기도 하는 데이터가 필요하다. 따라서 본 논문에서는 객체추적 분야에서 고전적으로 많이 사용되어온 PETS 09-S2L1[8] 데이터를 실험 데이터로 선정하였다.

그림 5. PETS 09-S2L1 데이터의 일부
Fig. 5. Part of PETS 09-S2L1 data

PETS 09-S2L1 데이터는 사람들이 서로 적절한 거리를 유지하고 있으면서 때로는 밀집했다가 엇갈려 지나가기도 하는 영상 데이터이다. 데이터에 대한 상세 정보는 표 2에 서술한다.

표 2. PETS 09-S2L1의 상세 정보
Table 2. Detail of PETS 09-S2L1 data

Resolution	768 × 576
FPS	7
Length	114 seconds
Number of frames	795
Number of annotated trajectories	19

본 실험에서 사용된 컴퓨터의 사양은 다음과 같다. CPU는 intel i7-8700K, 3.70GHz, GPU는 GTX 1080ti를 사용하였고 메모리는 32GB, 운영체제는 Ubuntu 18.04 LTS이다. 해당 환경에서 실험을 진행하였을 때 PETS 09-S2L1 데이터에 각각 기존 유클리드 거리를 이용한 방법, SORT를 이용한 방법 제안된 방법을 적용했을 때의 수행 시간 및 MOTA (Multiple Object Tracking Accuracy)는 표 3과 같다.

표 3을 통해 제안된 방법이 기존 유클리드 거리만을 사용한 방법의 MOTA가 73.4%인 것에 비해 제안된 방법을 통해 얻은 MOTA가 75%인 것을 보여 정확도를 개선시켰다는 것을 알 수 있다.

또한 수행 시간 측면에서 또한 기존 SORT에 비해 개선되었다는 것을 알 수 있다. 기존 SORT만 사용한 방법에 대해서는 수행 시간이 120.29초로 YOLO의 수행 시간인 105.6초를 감안했을 시 14.69초의 수행 시간을 갖는 반면, 제안된 방법의 수행 시간은 115.7초로 YOLO의 수행 시간을 제외했을 시 10.1초를 갖는다.

표 3. 각 추적 방법에 대한 실험 결과
Table 3. Experimental results for each tracking method

Result \ Tracking method	Euclidean distance	SORT	Proposed method
Execution time (including YOLO) (s)	105.69	120.29	115.7
Exucution time (excluding YOLO) (s)	0.09	14.69	10.1
MOTA(%)	73.4	77.4	75

따라서 제안된 방법이 더 적은 연산을 통해 수행 시간을 줄였음을 알 수 있다. 또한 SORT만 사용한 방법은 영상시간이 114초인 것에 비해 수행 시간이 120.29초로 딜레이가 발생했지만 제안된 방법을 사용했을 시 수행 시간은 영상의 길이와 일치하는 115.7초로 충분히 실시간 시스템에 적용할 수 있는 성능을 보였다.

제안된 방법이 SORT를 응용하였음에도 그보다 낮은 MOTA를 보이는 이유는 칼만 필터가 SORT만 사용한 방법에 비해 적은 정보를 얻었기 때문이다. 제안된 방법에서 SORT로 추적 방법이 전환되는 시점은 둘 이상의 객체가 객체의 높이의 절반보다 짧은 거리에 있을 때이다. 즉 두 객체가 교차할 때까지 칼만 필터가 많은 정보를 얻기는 어렵다. 추적 방법이 전환되는 거리를 제안된 방법대로 적용하였을 때와 그 두 배로 증가시켰을 때의 결과를 비교하면 그 사실을 알 수 있다.

표 4. 기준이 되는 거리에 따른 실험 결과
Table 4. Experimental results for the reference distance

Criteria to switch tracking method / Result	Height of object / 2	Height of object
Execution time (including YOLO) (s)	115.7	119.85
Exucution time (excluding YOLO) (s)	10.1	14.25
MOTA(%)	75	77

표 4에서는 추적 방법이 전환되는 거리가 변화했을 때 그에 따른 측정값의 변화를 확인할 수 있다. 기준이 되는 거리를 두 배로 증가시켰을 시의 MOTA는 77%로 SORT에 매우 가까운 수치이다. 즉 객체들이 제안된 방법에 비해 서로 더 멀리 떨어져 있어도 추적 방법이 SORT로 전환되기 때문에 칼만 필터는 더 많은 정보를 얻을 수 있어 MOTA가 증가하였다. 하지만 연산량 역시 증가했기 때문에 수행 시간 역시 SORT에 가깝게 증가한 것을 볼 수 있다. 따라서 원하는 성능을 얻기 위해서는 기준이 되는 거리를 적절하게 조절할 필요가 있다.

V. 결 론

본 논문에서는 다중 객체 추적 방법 중 높은 정확도를 갖지만 많은 연산을 필요로 하는 SORT와 적은 연산량을 갖지만 비교적 정확도가 낮은 유클리드 거리를 이용한 객체 추적법을 결합한 혼합형 저연산 다중 객체 추적 시스템을 제안하였다. 그 결과 유클리드 거리를 이용한 방법보다는 높은 정확도를 갖지만 SORT보다 수행 시간을 월등히 감소시켜 실시간 객체 추적까지도 가능하게 하는 시스템을 구현하였다. 그러나 수행 시간이 눈에 띄게 감소한 것이 비해 정확도는 크게 증가하지 않았다는 점의 한계점이 존재한다. 후속 연구에서는 정확도 또한 SORT에 비견할 정도의 증가량을 보이는 것을 목표로 한다.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, May 2016.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", Advances in neural information processing systems, pp. 91-99, Jun. 2015.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking", 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464-3468, Sep. 2016.

[4] R. Kalman, "A new approach to linear filtering and prediction problems a new approach to linear filtering and prediction problems", Transaction of the AMSE: Journal of Basic Engineering, Vol. 82, Issue 1, pp. 34-45, Mar. 1960.

[5] H. W. Kuhn, "The Hungarian method for the assignment problem", Naval Research Logistics Quarterly, Vol. 2, Issue1-2, pp. 83-97, Mar. 1955.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580-587, Jun. 2014.

[7] R. Girshick, "Fast R-CNN", The IEEE International Conference on Computer Vision (ICCV), pp. 1440-1448, Dec. 2015.

[8] J. Ferryman and A. Shahrokni, "PETS 2009: Dataset and challenge", Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveillance, pp. 1-6, Dec. 2009.

저자소개

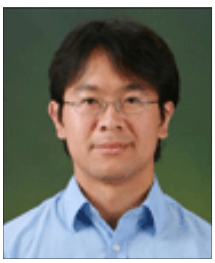
우 윤 희 (Yun-Hee Woo)

2017년 8월 : 한림대학교 컴퓨터공학과(공학사)
2019년 2월 : 한림대학교 컴퓨터공학과(공학석사)
2019년 3월 ~ 현재 : 한림대 컴퓨터공학과 박사과정
관심분야 : 머신러닝, 딥러닝, 객체 인식, 최적화 알고리즘

최 권 택 (Kwon-Taeg Choi)

2006년 2월 : 한림대학교 컴퓨터공학과(공학석사)
2011년 2월 : 연세대학교 컴퓨터공학과(공학박사)
2016년 3월 ~ 현재 : 강남대학교 소프트웨어응용학부 교수
관심분야 : 가상현실, 증강현실, 모바일컴퓨팅, 기계학습, HCI

이 정 근 (Jeong-Gun Lee)



1996년 2월 : 한림대학교 전자계산학과(이학사)
1998년 2월 : 광주과학기술원 정보통신공학과(공학석사)
2005년 2월 : 광주과학기술원 정보통신공학과(공학박사)
2005년 5월 ~ 2007년 7월 : 캠브리지 대학교 컴퓨터 랩, 박사후 연구원
2014년 2월 ~ 2014년 12월 : 캠브리지 대학교 컴퓨터 랩, 방문 연구원
2008년 2월 ~ 현재 : 한림대학교 컴퓨터공학과 교수
관심분야 : VLSI/CAD, 컴퓨터구조, 비동기 회로 설계, 디지털 회로설계, 병렬 컴퓨팅, GPU 구조