일반논문 (Regular Paper)
방송공학회논문지 제23권 제5호, 2018년 9월 (JBE Vol. 23, No. 5, September 2018)
https://doi.org/10.5909/JBE.2018.23.5.642
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

점유 센서를 위한 합성곱 신경망과 자기 조직화 지도를 활용한 온라인 사람 추적

길 종 인[a)], 김 만 배[a)‡]

Online Human Tracking Based on Convolutional Neural Network and Self Organizing Map for Occupancy Sensors

Jong In Gil[a)] and Manbae Kim[a)‡]

요 약

빌딩, 집에 설치되어 있는 점유 센서는 사람이 없으면 소등하고, 반대이면 점등한다. 현재는 주요 센서로 PIR(pyroelectric infra-red)이 널리 사용되고 있다. 최근에 비전 카메라 센서를 이용하여 사람 점유를 검출하는 연구가 진행되고 있다. 카메라 센서는 정지된 사람을 검출할 수 없는 PIR의 단점을 극복할 수 있는 장점이 있다. 이동 및 정지된 사람의 추적은 카메라 점유 센서의 주요 기능이다. 본 논문에서는 합성곱 신경망 모델과 자기 조직화 지도를 활용한 온라인 사람 추적 기법을 제안한다. 오프라인에 모델을 학습시키기 위해서는 많은 수의 훈련 샘플이 필요하다. 이러한 문제를 해결하기 위해, 학습되지 않은 모델을 사용하고, 실험 영상으로부터 직접 훈련 샘플을 수집하여 모델을 갱신한다. 오버헤드 카메라로 실내에서 촬영한 영상을 이용하여, 제안 방법이 효과적으로 사람을 추적하고 있음을 실험을 통해 증명하였다.

Abstract

Occupancy sensors installed in buildings and households turn off the light if the space is vacant. Currently PIR(pyroelectric infra-red) motion sensors have been utilized. Recently, the researches using camera sensors have been carried out in order to overcome the demerit of PIR that cannot detect stationary people. The detection of moving and stationary people is a main functionality of the occupancy sensors. In this paper, we propose an on-line human occupancy tracking method using convolutional neural network (CNN) and self-organizing map. It is well known that a large number of training samples are needed to train the model offline. To solve this problem, we use an untrained model and update the model by collecting training samples online directly from the test sequences. Using videos capurted from an overhead camera, experiments have validated that the proposed method effectively tracks human.

Keyword : on-line tracking, convolutional neural network, self organizing map, occupancy sensor

Ⅰ. 서 론

현재 많은 건물에는 사람이 존재하면 점등하고(light on), 사람이 없으면 소등하는(light off) 점유 센서(occupancy sensor, motion sensor)가 설치되어 있다[1,2]. 전기소비를 줄이기 위해 설치된 것으로 대부분의 점유 센서는 현재 PIR (pyroelectric infra-red)을 사용하고 있다. PIR은 사람의 신체에서 발생하는 열(thermal temperature)을 감지하고, 열의 변화량을 계산하여 움직임을 측정한다.

최근에 PIR 대신에 카메라 비전 센서를 활용하려는 연구가 진행되고 있다[3~7]. 카메라 센서는 위에서 언급한 PIR의 단점을 극복할 수 있고, 부가적으로 사람 추적, 사람 명수 파악, 사람 행위 등 다양한 지능 정보의 획득이 가능하다. 비전 센서는 사람의 추적이 중요한 기능이고, 이 추적 기술은 장시간 정지되어 있는 사람을 구별하는 역할을 담당한다. 장시간 정지된 사람의 미검출은 PIR의 단점이다.

Benezeth 등은 CAPTHOM 프로젝트의 일환으로 비전 센서를 활용하여 사람 점유 및 행위를 분석하는 연구를 수행하였다[3]. 이를 위해 얼굴 검출 및 추적 기술을 활용하였다. 측면에 카메라를 고정하였고, 최대 2명의 사람에 대해서 실행하였다. Han 등은 비전 센서와 PIR를 이용하여 사람 검출 연구를 수행하였다[4]. Nakashima 등은 카메라 센서를 활용하여 움직이는 사람을 검출하는 방법을 제안하였다[5]. 한 명에 대한 실험이어서, 여러 사람이 있는 경우에는 적용의 어려움이 있다. Amin 등은 PIR과 비전 센서를 이용하여 사람의 명수를 구하는 연구를 진행하였다[6]. 많은 연구에서는 PIR과 비전 센서를 동시에 사용하여 성능을 높이는데, 단점은 2개 이상의 이종 센서를 사용하는 것이다. [3,5]에서는 카메라만 사용하여 점유를 검출하는 기법을 제안하지만, 사람의 가려짐이 발생할 때에 사람 검출의 성능이 저하되고, 특히 장시간 정지된 사람에 대한 처리 방법이 부족하다. 이를 위해 Gil 등은 비전 센서만을 이용하여 점유를 검출하는 기법을 제안하였다[7]. Gil의 연구에서는 실내 공간에서 사람이 입실하면 점등하고, 모든 사람이 퇴실하면 소등되도록 설계하였다. 추적 기법으로는 MHI(Motion History Image)를 이용하여 이동하는 사람을 추적하였다.

본 논문에서는 비전 센서 기반 사람 추적 기법을 제안한다. 카메라가 설치된 공간에는 많은 사람들이 입실 및 퇴실이 반복되고, 의자, 소파, 책상 등 다양한 가구들이 공존하고 있다. 다양한 특성을 갖는 객체들로부터 목표 사람만을 정확하게 추적하는 것이 본 연구의 목적이다. 또한 제안 방법은 점유센서의 실시간 처리를 만족하기 위해서 온라인 트랙킹으로 설계된다.

최근에는 객체 추적을 위해 검출 기법을 활용한 방법이 많이 연구되고 있는데, 이러한 방법을 검출에 의한 추적(Tracking-by-Detection: TbD)이라 한다. 객체를 검출하기 위한 검출기를 학습하기 위해 여러 가지 기계학습법이 사용될 수 있다. 이러한 TbD 기법은 모델을 사전에 학습하는 방법과 사전에 학습되지 않은 모델을 사용하는 방법으로 구분된다. 사전에 학습된 모델을 사용할 경우, 모델의 학습을 위해 많은 수의 훈련 샘플이 필요하고, 이는 큰 비용을 초래한다. 비록 충분한 양의 훈련 샘플이 충족되었다 할지라도, 높은 정확도를 갖는 분류기의 학습을 위해 많은 시간이 필요하다. 그러나 일단 많은 수의 훈련 샘플이 확보가 된다면 해당 모델은 범용적인 목적으로 사용될 수 있다.

모델을 사전에 학습하지 않는 경우에는, 사전에 훈련 샘플을 준비하지 않아도 된다는 장점이 있다. 이는 실용성이 크므로 큰 장점이 될 수 있다. 추적을 수행하면서 순차적으로 훈련 샘플을 획득하고 모델을 업데이트한다. 이러한 방법은 비교적 최근에 획득한 훈련샘플에 과적합(overfitting)될 가능성이 있다. 그러나 이러한 문제는 객체의 추적에 있어서는 비교적 큰 문제가 되지 않는다.

최근에는 현재 컴퓨터비전 분야에서 가장 널리 활용되는

a) 강원대학교 컴퓨터정보통신공학과(Dept. of Computer and Communications Engineering, Kangwon National University)
‡ Corresponding Author : 김만배(Manbae Kim)
E-mail: manbae@kangwon.ac.kr
Tel: +82-33-250-6395
ORCID:http://orcid.org/0000-0002-4702-8276
※ 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터지원사업의 연구결과 (IITP-2018-0-01433) 및 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2017R1D1A3 B03028806).
※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-0-01433) supervised by the IITP(Institute for Information & communications Technology Promotion) and This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2017R1D1A3B03028806).
· Manuscript received July 6, 2018; Revised August 9, 2018; Accepted, August 9,2018.

합성곱 신경망(Convolutional Neural Network: CNN)을 활용한 온라인 객체 추적 기법이 연구되고 있다[8-10]. CNN은 객체 검출 및 인식에 탁월한 성능을 보여주는 것으로 알려져 있다. 이러한 방법들은 모두 TbD 메커니즘을 적용하고 있다. CNN은 객체 검출 및 인식에 탁월한 성능을 보여주는 것으로 알려져 있다. 탐색 영역을 모두 탐색하는 비용을 줄이기 위해서 자기 조직화 지도(Self Organizing Map)을 이용하여 탐색 범위를 줄이는 기법을 사용한다.

본 논문의 구성은 다음과 같다. 다음 장에서는 제안하는 방법의 전체적인 흐름을 설명하고, 신경망 모델의 학습을 위한 훈련 집합을 추출하는 방법에 대하여 설명한다. Ⅲ장에서는 제안하는 객체 추적 시스템의 필수 요소인 합성곱 신경망 모델에 대해서 자세히 설명한다. 실험 결과는 Ⅳ장에서 정리하고, 마지막으로 Ⅴ장에서 결론짓는다.

Ⅱ. 온라인 객체 추적

1. 추적 시스템의 개요

그림 1은 제안 방법의 전체 흐름도를 보여준다. 객체 추적은 미래에 입력될 프레임으로부터 두 가지 요소를 추정하는 과정이다. 첫 번째는 객체의 위치이고 두 번째는 객체의 크기이다. 먼저 객체의 위치를 결정하기 위해서는 객체가 존재할 수 있는 모든 후보 위치로부터 후보 이미지 패치(patch)들을 추출한다. 이때 후보 이미지 패치의 너비와 높이를 각각 B_h, B_w라 하고, 전체 이미지 크기를 I_h, I_w라 하자. 또한 객체의 가능한 너비와 높이의 변동을 $\pm\alpha$, 즉 너비가 $[B_h-\alpha, B_h+\alpha]$, 높이가 $[B_w-\alpha, B_w+\alpha]$의 범위 내에 존재한다고 가정할 때, 이미지 패치의 크기가 가질 수 있는 경우의 수는 모두 $(4\times\alpha+2)$개가 존재한다. 따라서 가능한 후보 이미지 패치의 수는 총 $(I_h-B_h+1)\times(I_w-B_w+1)\times(4\times\alpha+2)$개이다. 객체의 탐색범위를 한정한다고 하여도 여전히 가능한 이미지 패치의 수는 크다. 예를 들어, B_h, B_w를 각각 32라 하고, 탐색 범위의 너비와 높이를 각각 200, α를 10이라 하자. 이럴 경우 검사해야 할 후보 이미지 패치의 수는 모두 32,592개이다. 탐색 범위와 객체의 크기를 작게 한정하였음에도 이미지 패치를 모두 검사해야 한다. 크기가 커질수록 검사해야 할 이미지 패치의 수는 기하급수적으로 증가한다. 이는 시스템에 많은 부담을 주게 된다.

따라서 모든 후보 이미지 패치들을 검사하기 전에 후보 이미지 패치들을 탐색하는 작업의 선행이 필요하다. 이를 위해 자기 조직화 지도(Self Organizing Map: SOM)을 활용한다.

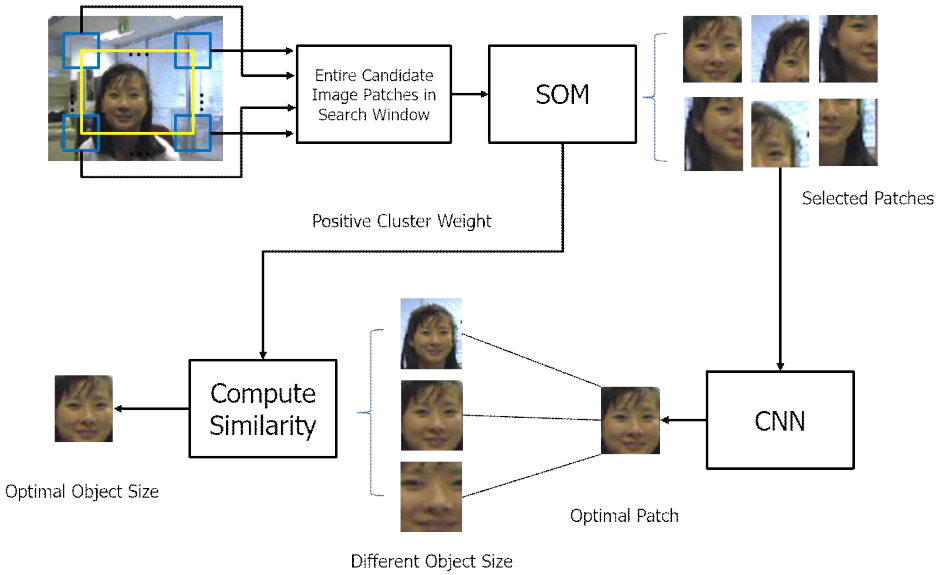

그림 1. 객체의 위치 및 크기의 결정을 위한 시스템의 전체 흐름도
Fig. 1. Overall flow diagram of proposed system for determining the location and scale of a target object

SOM은 비지도 학습을 사용하는 신경망의 일종으로서, 주로 군집화를 수행하기 위한 목적으로 개발되었다[11]. 먼저 탐색 범위로부터 후보 패치들을 추출한다. 이때 객체의 크기는 고려하지 않는다. 추출된 후보 패치들은 SOM으로 입력되고, SOM으로부터 목표 객체일 확률이 높은 패치들을 선택한다.

선별된 후보 패치는 CNN으로 입력된다. 이 단계에서 하나의 패치만을 선택하게 된다. 하나의 패치가 선택되었다는 것은 객체의 위치가 결정되었음을 의미한다. 객체 위치가 결정되었다면, 다음 단계는 객체 크기를 결정하는 단계가 진행되어야 한다. 결정된 객체의 위치로부터 다양한 크기를 가지는 후보 패치를 다시 추출한다. 즉, 동일한 위치에서 크기만 달리하여 후보 패치를 생성한다. 앞선 단계에서 훈련된 SOM으로부터 앞서 추출한 크기가 다른 후보 패치들을 다시 검사한다. 그 중 가장 가능성이 높다고 판단되는 크기를 갖는 패치를 최적의 패치로 선택한다. 이 과정으로부터 객체의 최적 위치와 크기를 결정한다.

2. 훈련 샘플 수집

제안 방법은 영상으로부터 훈련 샘플을 직접 수집하는 온라인 추적 기술이다. 수집된 훈련샘플은 모델을 학습하는데 이용된다. 이러한 방법은 많은 장점을 가질 수 있다. 먼저 사전에 훈련 샘플을 이용하여 검출기를 학습하는 경우, 훈련샘플과 실험영상에 많은 차이가 있을 때 (예를 들어, 해상도 및 화질의 차이) 충분한 성능을 내지 못할 가능성이 크다. 그러나 실험영상에서 훈련 샘플을 직접 수집하게 되면 이러한 차이로부터 발생하는 성능의 차이를 극복할 수 있다. 또한, 검출기의 사전 훈련을 위해서는 많은 수의 훈련 샘플이 필요하다. 이렇게 많은 수의 샘플을 생성하는 것은 많은 비용이 필요하다. 이러한 문제는 실험영상으로부터 온라인으로 훈련샘플을 수집함으로써 이러한 문제의 해결이 가능하다.

제안방법에서 객체 추적을 위한 초기화는 필요하지 않다. 단 첫 프레임에서 추적될 객체의 위치는 알려져 있다고 가정한다. 해당 객체의 위치를 (x,y), 객체의 높이와 너비를 (w,h)이라 할 때, 해당 위치를 중심으로 바운딩 박스를 설정한다. 또한 획득한 이미지 패치에 좌우 반전을 수행함으로써 두 개의 긍정(positive) 패치를 획득한다.

추적 객체의 위치로부터 상하좌우 네 방향에 대해 동일한 크기의 바운딩 박스를 취한다. 즉, 네 위치의 좌표는

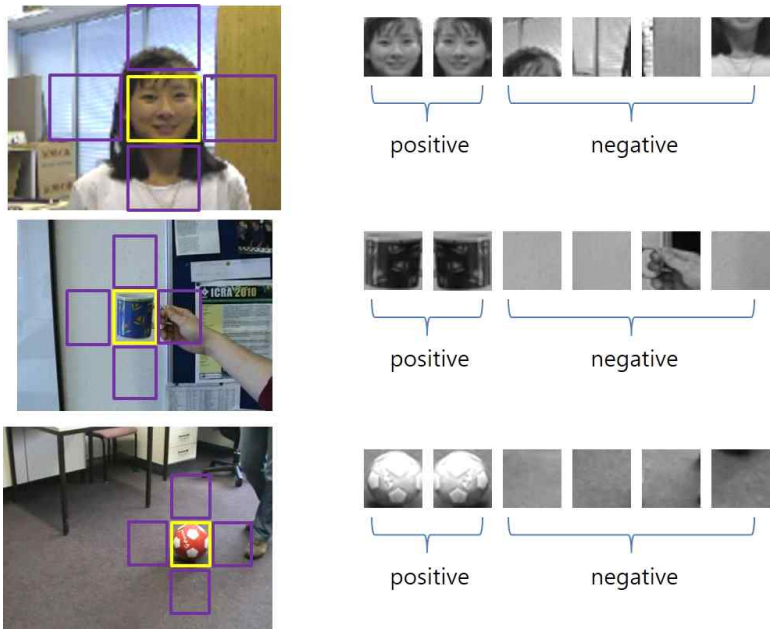

그림 2. 추적되는 객체로부터 얻어지는 긍정 및 부정 바운딩 박스 (적색: 긍정 이미지 패치, 청색: 부정 이미지 패치)
Fig. 2. Positive and negative bounding boxes obtained from a tracked object. (red: positive image patch, blue: negative image patch)

$(x \pm w, y \pm h)$가 된다. 이 위치에서 동일한 크기의 바운딩 박스를 생성하고, 이미지 패치의 클래스를 부정(negative)으로 설정한다. 이로써, 총 2개의 긍정 샘플과 4개의 부정 샘플을 획득할 수 있다. 그림 2와 같이 추적 객체가 영상 내부에 존재하는 동안 매 프레임마다 6개의 훈련 샘플을 획득한다.

3. 자기 조직화 지도

자기 조직화 지도(Self Organizing Map: SOM)은 비지도 학습을 사용하는 신경망의 한 종류로써 주로 군집화를 수행하기 위한 목적으로 개발되었다. 비지도 학습이므로 각각의 샘플은 부류가 정해져 있지 않다. 이는 K-평균 알고리즘의 온라인 방식이며, Kohonen이 개발하였기 때문에 Kohonen 네트워크라고도 불린다. 이 모델은 마치 하나의 그룹이 승리하면 업데이트 되고 다른 그룹은 전혀 업데이트 되지 않기 때문에, 승자 독식법(winner-take-all)이라고도 한다. SOM은 기본적으로 입력층과 경쟁층으로 구성된 2-계층을 갖는 신경망이다. SOM 학습의 목적은 네트워크의 특정 부분이 특정 입력 패턴과 유사하게 반응하도록 하는 것이다. 입력층의 노드 수는 입력 특징 벡터의 차원과 동일하고, 경쟁층의 노드 수는 군집화하려는 클래스의 수와 일치한다. 경쟁층의 노드들이 군집을 형성하기 위해 서로 경쟁하기 때문에 경쟁층이라 한다. 경쟁층에 M개의 노드가 있다고 가정하면 최종적으로 생성되는 군집의 수는 군집을 생성하지 못한 노드도 발생할 수 있기 때문에 M개 이하이다. 즉 노드의 수 M은 최대 군집 개수이다.

본 장에서는 SOM을 객체 추적에 맞도록 변형한 방법을 제안한다. 제안 기법은 객체의 현재 위치로부터 1개의 긍정 샘플(1개를 좌우 반전하여 2개를 생성하므로 1개로 봐도 무방)과 4개의 부정 샘플을 획득하여 훈련에 사용한다. 탐색 공간에 존재하는 모든 후보 패치 집합에 군집화를 수행할 때, 획득한 5개의 훈련 샘플을 각 군집들의 중심으로 설정하여 부정 샘플에 해당하는 군집들을 제거할 수 있다면 테스트해야 할 후보 이미지들이 상당수 제거 될 것이다. 이를 위해 SOM에서 경쟁 층의 노드 수를 5개로 한정하고 초기 가중치 값을 5개의 훈련 샘플로부터 초기화하면 된다. 경쟁 층의 노드 수가 5개에 불과하므로 경쟁층은 1차원으로 배열한다. 입력 노드의 수는 입력된 이미지 패치 픽셀의 수이고, 경쟁 노드의 수는 5개이므로 가중치는 총 $5 \times N$개다. 이때, N은 입력 특징 벡터의 차원 수이다. 가중치의 수가 상당히 적기 때문에 SOM 학습에 소요되는 시간도 짧다. 또한 최대 반복 횟수를 1로 두어서 한 번만 반복하도록 한다. 반복 횟수를 1로 설정하는 이유는, 실제로 연속된 프레임에서 객체의 외형의 변화는 심하지 않기 때문이다. 또한 SOM을 활용하는 가장 큰 목적은 후보 이미지 패치 선별인데, 반복 횟수를 높게 설정하면 연산량이 늘어나게 되므로 SOM을 활용하는 의미가 없어지게 된다. 이는 다시 긍정 노드의 가중치 벡터의 값이 심하게 변하면 안된다는 의미이다.

기존에 여러 가지 군집화 알고리즘이 존재하지만 SOM 알고리즘을 사용하는 이유도 이와 유사하다. 추적 객체의 외형이 연속된 프레임에서 분명히 변화하지만, 그 변화의 정도가 크기 않기 때문에 사용된 모델의 변화 역시 크지 않아야 한다. 예를 들어 K-Means 알고리즘은 각 샘플이 어떤 군집에 포함되는지를 먼저 검사한 후에, 각 군집에 속한 샘플들로부터 평균을 측정하여 군집의 중심을 평균점으로 이동하는 과정을 반복적으로 수행한다. 또한 이 중심점의 양은 상대적으로 클 수 있다. 이로부터 가중치 벡터값의 변화를 둔감시킴으로써 객체의 외형을 유지하는 것이 가능하다. 다음은 객체 추적을 위해 후보를 선별하기 위한 SOM 알고리즘을 보여준다.

입력: 샘플 집합 $X=\{x_1, x_2, \cdots, x_T\}$, 최대 군집 개수 M, 학습률 α

출력: 긍정 후보 이미지 패치 집합 C

1. SOM의 모든 노드 가중치 벡터에 5개의 훈련 샘플로부터 얻은 특징값을 할당
2. 탐색 범위 내의 각 위치로부터 후보 이미지 패치 x_t를 차례대로 추출
3. Euclidean distance를 이용하여 입력 벡터와 노드 가중치 벡터 사이의 유사도를 측정
4. 가장 작은 거리를 갖는 노드를 승자노드로 선택
5. 승자노드와 이웃한 노드의 가중치 벡터를 다음 식을 이용하여 갱신

$W_m^{\neq w} = W_m^{old} + \alpha \cdot (x_t - W_m^{old})$

6. t를 1증가 시키고, t가 T가 아니라면 2번으로 돌아감
7. 샘플 집합 X의 모든 x_t에 대해 가장 가까운 벡터 W_m를 찾아서 x_t를 c_m에 배정
8. 군집 $c_m, m = 1,...,M$중에 긍정 샘플 집합에 해당하는 것을 선택하여 반환

SOM을 이용하여 후보 이미지 패치 집합을 생성하였다면, 해당 집합의 이미지 패치가 CNN을 이용하여 평가되고, 그로부터 출력되는 점수들을 비교하여 객체의 최종 위치를 결정한다. 이렇게 위치가 결정되었다면, 다음 단계는 스케일을 결정해야 한다. 마찬가지로 SOM을 이용하여 스케일을 결정할 수 있는데, 이전 프레임에서 객체의 크기를 기준으로 하여, 객체의 가로, 세로 크기를 0.5배부터 1.5배까지 0.1의 간격 크기를 갖도록 하여 객체의 크기를 달리하여 후보 스케일 이미지 패치들을 얻는다. 즉, 총 11개의 후보 스케일 이미지 패치를 얻을 수 있다. SOM에서 사용된 가중치를 이용하여 유클리디안 거리(Euclidean distance)를 측정하여 가장 가까운 거리를 갖는 후보 스케일 이미지 패치를 선택하도록 한다.

Ⅲ. 합성곱 신경망 모델

다음 그림 3은 객체 추적을 위한 CNN의 구조를 보여준다. CNN은 두 개의 합성곱층(convolutional layer)과 두 개의 완전결합층(fully-connected layer)으로 구성된다. 활성함수는 시그모이드(sigmoid)이고, 차원 축소를 위해 평균값 풀링이 사용된다. 획득한 이미지 패치는 32x32x3의 크기로 변경되어 입력된다. 첫 번째와 두 번째 합성곱에서는 각각 3x3의 크기를 갖는 필터가 8개씩 존재한다. 패치에 콘볼루션이 적용되면 노드의 수는 30x30x8이 되고, 392개의 노드가 완전결합층에 입력된다. 완전결합 층에서는 두 개의 은닉층(hidden layer)이 존재하고 각각의 층에는 100, 80개의 노드가 있다. 완전결합층의 마지막에는 활성함수로서 소프트맥스(softmax)가 사용되고, 모멘텀을 적용하였다.

1. CNN 모델 학습

CNN 모델로부터 현재 프레임의 객체 위치가 결정되면, 새로운 훈련샘플을 수집하여 CNN 모델을 갱신한다. 수집된 샘플의 신뢰도는 현재 CNN 모델의 정확성에 따라 결정된다. 만일 CNN 기반의 추적기가 현재 프레임에서의 객체의 위치를 올바르게 추적하지 못했다면, 훈련 샘플도 올바르게 수집되지 못할 것이다. 본 논문에서는 이러한 문제를 해결하기 위해 컬러 모델을 이용한다. 먼저 프레임 $t-1$에서 객체의 위치가 올바르게 추적되었다고 가정했을 때, 해당 패치로부터 컬러 히스토그램을 구한다. 히스토그램을 사각형 형태의 패치로부터 직접 생성할 경우, 배경이 히스토그램에 영향을 미칠 수 있다. 따라서 사각 바운딩 박스 대신 타원 형태로 경계를 설정하여 히스토그램을 생성한다. 다음으로 프레임 t에서 획득한 히스토그램을 역투영하여 확률맵(probability map) BP를 획득한다. 그림 4는 확률맵과 역확률맵을 보여주고 있다.

BP로부터 적분확률맵(Integral Probability Map) P는 다음 식에서 구해진다.

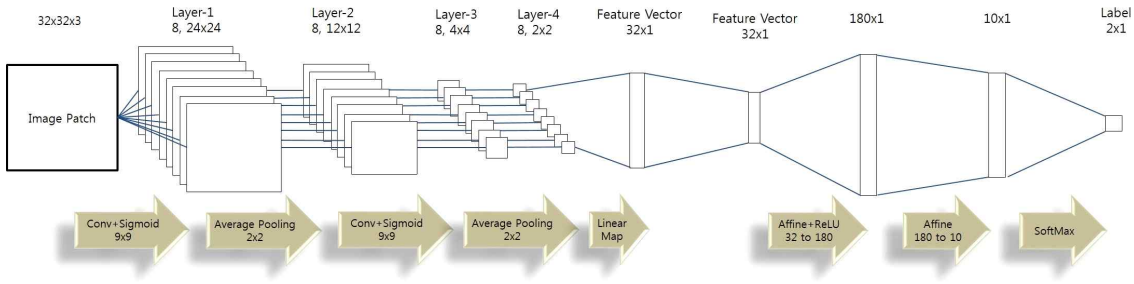

그림 3. 추적을 위한 CNN의 구조
Fig. 3. Architecture of CNN for visual tracking

$$P_p(x_n,y_n)=\sum_{a=0}^{h_n-1}\sum_{b=0}^{w_n-1}BP_p(x_n+a,y_n+b)$$
$$P_n(x_n,y_n)=\sum_{a=0}^{h_n-1}\sum_{b=0}^{w_n-1}BP_n(x_n+a,y_n+b) \quad (1)$$

여기서 BP_p는 확률맵, BP_n은 역확률맵이다. P_p는 BP_p의 합으로 얻어진 적분확률맵이고, P_n은 역적분확률맵이다.

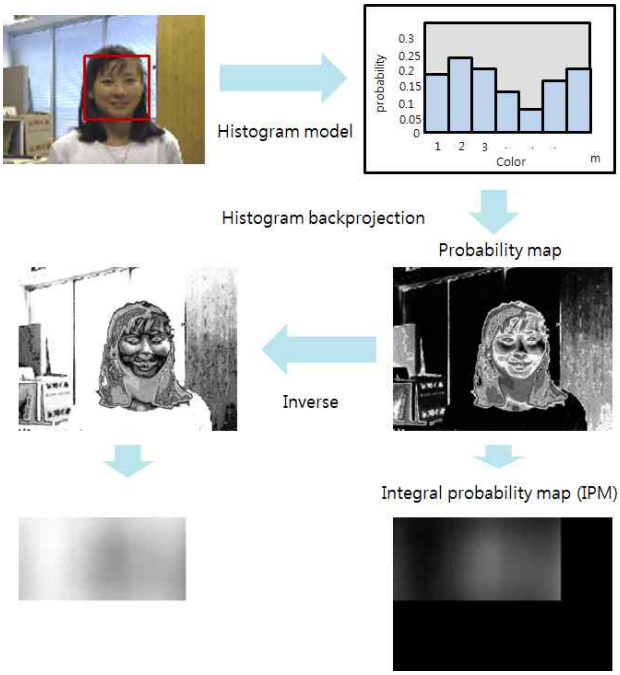

그림 4. 적분 확률맵 생성 과정
Fig. 4. Procedure of generating Integral Probability Map

그림 4와 같이 확률맵에서 나타나는 픽셀 값은 샘플의 레이블이 긍정일 가능성으로 해석할 수 있다. 반대로 그의 역확률맵은 훈련 샘플이 부정 패치일 가능성을 나타낸다고 볼 수 있다. 확률맵은 식 (1)을 이용하여 확률맵의 이미지 패치 내부의 모든 픽셀 값을 합산한 적분확률맵으로 변환한다. 적분확률맵의 특성을 이용하여 새로운 손실함수를 정의한다. 식 (2)는 Cross entropy 손실 함수이다.

$$J(l_n,d_n)=\sum_{i=1}^{M}\{-d_{n,i}\ln(l_{n,i})-(1-d_{n,i})\ln(1-l_{n,i})\} \quad (2)$$

여기서 ln은 자연로그이다. M은 출력 노드의 수이고, i는 출력 노드의 인덱스이다. 여기에서는 2-class만 존재하므로 $M=2$이다. 또한, n은 훈련 샘플의 인덱스이다. d_n은 훈련 샘플의 레이블(ground truth), l_n은 CNN의 출력 벡터이다. 식 (1)에서 획득한 확률값을 이용하여 식 (3)의 새로운 손실 함수를 정의한다. 이렇게 생성된 식 (2)와 (3)을 결합하여 식 (4)의 최종 손실 함수를 정의한다.

$$C(l_n,d_n)=\frac{1}{N}\sum_{n=1}^{N}\{d_nP_p(x_n,y_n)+(1-d_n)(1-P_n(x_n,y_n))\} \quad (3)$$
$$L(l_n,d_n)=C(l_n,d_n)\times J(l_n,d_n) \quad (4)$$

여기서 N은 훈련 패치의 수이고, 실험에서는 $N=6$으로 설정하였다. (x_n,y_n)은 n번째 훈련 패치의 위치, (w_n,h_n)은 너비와 높이이다.

이미지 패치의 레이블이 긍정일 경우, 해당 패치가 객체의 컬러모델과 유사한 색상 분포를 가지게 되면 CNN 모델 갱신에 있어서 상대적으로 높은 가중치를 얻게 된다. 만일 어떤 패치가 객체의 컬러 모델과 유사하지 않다면, 그것은 추적하고자 하는 객체일 가능성이 현저히 낮아진다. 따라서 해당 패치의 레이블이 부정일 경우, CNN 모델 갱신에 있어서 높은 가중치를 얻게 된다. 즉, 긍정 패치이지만 컬러모델과 큰 차이를 갖거나, 부정 패치이지만 컬러 모델과 유사한 경우에는 모호한 경우로 규정하고 낮은 가중치를 할당함으로써 CNN 모델 갱신의 오류를 줄이고자 하였다.

그러나 식 (4)의 손실 함수를 이용하여 CNN 모델을 학습하여도 학습에 사용된 훈련 패치의 수는 6개이므로 적은 수의 훈련으로는 CNN 모델이 원하는 값으로 수렴될 수 없다. 이러한 이유로 인해, 일반적으로 CNN 모델을 학습시킬 때, 전체 학습 데이터에 대해서 한번만 학습시키는 것이 아니라 재학습을 시키게 된다. 이렇게 전체 학습 데이터를 한 번씩 모두 학습시킨 횟수를 에폭(epoch)이라 한다. 보통 에폭을 고정시켜놓고 학습을 진행한다. 앞선 과정을 통해 훈련샘플을 수집한 후에, 전체 데이터를 한 번씩 학습할 때마다 식 (5)을 이용하여 훈련 오차를 계산한다.

$$e=\frac{1}{N}\sum(l_n-d_n)^2 \quad (5)$$

이러한 오차가 임계치보다 작을 때까지 훈련을 반복한다. 즉, 프레임 t에서 객체의 위치를 예측하고, 이로부터 훈련샘플을 수집하여 모델을 학습한 후, 훈련에 사용된 훈련샘플을 이용하여 모델을 테스트한다. 훈련샘플로부터 모델의 오차가 수렴할 때까지 모델의 훈련을 반복한다.

2. 객체 추적

새로운 프레임이 입력되면 학습된 CNN 모델을 이용하여 추적하고자 하는 객체의 위치를 예측한다. 이전 프레임에서 객체의 위치를 중심으로 탐색 범위를 설정한다. 탐색 범위의 수평 길이를 s_h, 수직 범위를 s_v라 할 때, 해당 탐색 범위에서 총 $s_h \times s_v$개의 후보 이미지 패치를 획득할 수 있다. 획득한 모든 패치에 대해 학습된 CNN 모델로 입력되면, 2D 벡터의 형태로 출력이 된다. 출력된 벡터는 $v=\{s_p,s_n\}$의 형태를 갖는다. s_p는 긍정 확률, s_n은 부정 확률이다. 출력에 소프트맥스를 적용하였으므로, $s_p+s_n=1.0$이다. 따라서 결국 s_p가 해당 패치가 긍정일 확률이라고 규정할 수 있다. 여기에서는 s_p를 점수로 규정하여, 가장 높은 점수를 갖는 곳에 객체가 위치해있다고 판단할 수 있다.

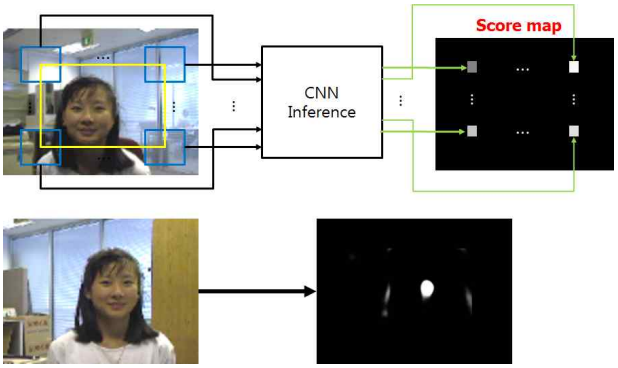

그림 5. 객체 탐색 영역(노란색)과 후보 이미지 패치(청색)
Fig. 5. Object search range (yellow) and candidate image patch (blue)

모든 후보 이미지 패치에 대해 점수를 획득하고, 이를 영상의 픽셀에 할당하면 점수맵(score map)을 획득할 수 있다. 이를 그림 5에서 보여주고 있다. 탐색 범위를 노란색으로 나타내었고, 개별 이미지 패치들을 파란색으로 표현하였다. 첫 프레임에서는 모델이 학습되어있지 않기 때문에 점수맵을 획득할 수 없다. 학습된 CNN 모델은 두 번째 프레임에서부터 적용이 가능하므로, $t=2$부터 점수를 획득할 수 있다. 그림 6에서 보는 바와 같이 추적하고자 하는 객체의 근처에서 높은 점수가 나타남을 확인할 수 있다. 그러나 1.0의 점수를 갖는 위치는 많이 존재한다. 따라서 점수맵의 무게 중심을 예측된 위치로 판단하였다. 무게 중심은 다음과 같이 계산된다.

$$m_{pq}=\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} i^p j^q f(i,j) \tag{6}$$

$$x_c=\frac{m_{10}}{m_{00}}, y_c=\frac{m_{01}}{m_{00}} \tag{7}$$

$t=2$일 때, 객체의 위치를 찾았다면, 그 위치로부터 다시 긍정샘플과 부정샘플을 앞서 기술한 바와 동일하게 획득한다. 이로부터 CNN 모델을 갱신하고, $t=3$일 때의 점수맵을 획득하여 객체의 위치를 예측한다. 이와 같은 과정을 반복함으로써 연속적으로 객체의 위치를 추적하게 된다.

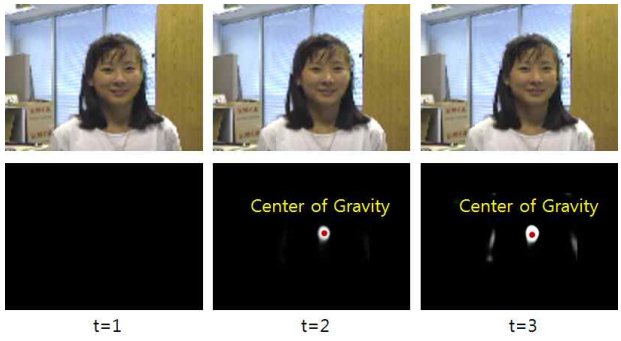

그림 6. 입력 영상과 점수맵
Fig. 6. Input images and score maps

Ⅳ. 실험 결과

실험을 위해 총 10개의 영상을 카메라로부터 직접 획득하였다. 영상 해상도는 1024x786이고 RGB 입력이다. 카메라는 연구실의 천장에 수직으로 설치하였고, 천장의 높이는 2.7m이다. 총 10개의 실험 영상에 대하여 제안 방법을

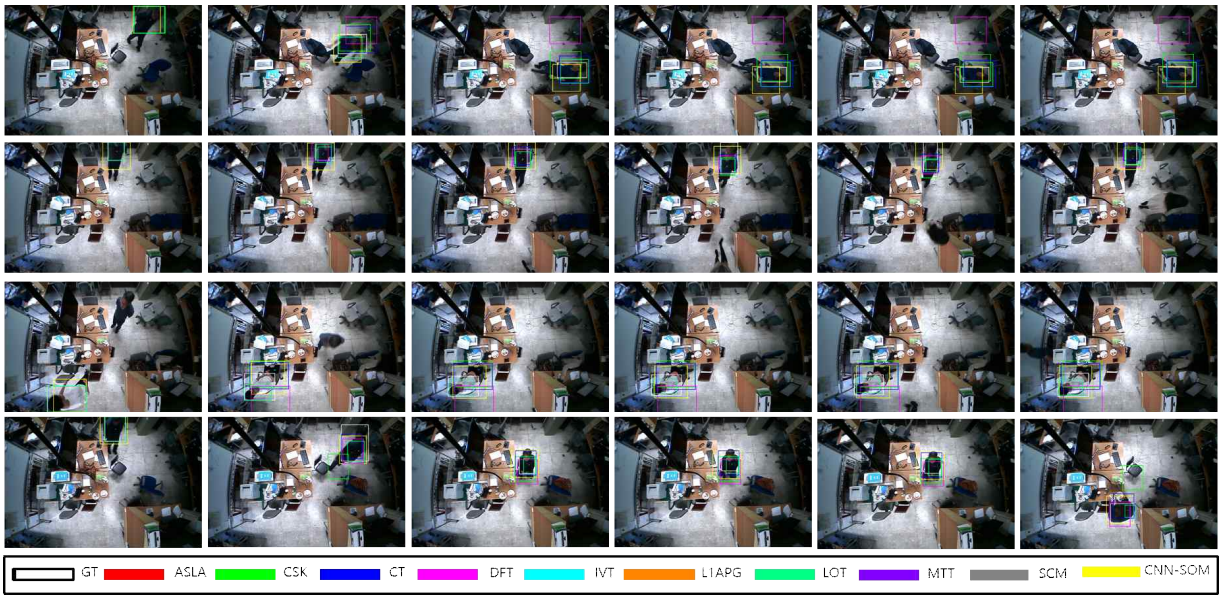

그림 7. 4개의 실험 영상에 대한 9개의 추적기를 적용한 실험 결과. 제안 방법은 CNN-SOM임
Fig. 7. Qualitative results of the 9 trackers over 4 test sequences. CNN-SOM is the proposed method

적용하였고, 그중 4개의 결과를 그림 7에서 보여주고 있다. 1행부터 4행까지 차례대로 Video1, Video8, Video9, Video 10에 대한 결과이다.

타 추적 알고리즘과 비교를 위해 9개의 알고리즘을 적용하였고, 각기 다른 색으로 표현하였다. 9개의 알고리즘은 순서대로 Adaptive Structural Local Sparse Model(ASLA)[12], Circulant Structure of Tracking-by-Detection with Kernel(CSK)[13], Compressive Tracking(CT)[14], Distribution Fields for Tracking(DFT)[15], Incremental Learning for Robust Visual Tracking(IVT)[16]. L1 Tracking using Accelerated Proximal Gradient Approach(L1APG)[17], Locally Orderless Tracking(LOT)[18], Multi-task Sparse Learning(MTT)[19], Sparsity-based Collaborative Model (SCM)[20]이다. 제안방법은 CNN-SOM이다.

정량적 평가를 위해 추정된 결과가 두 가지 객관적 측정 도구를 이용하여 평가하였다. 정량적 평가를 위한 실측 정보(Ground Truth)를 실험 영상으로부터 직접 추출하였다. 추적하고자 하는 객체는 모두 사람인데, 사무실, 연구실, 회의실 등과 같은 장소의 경우 대부분 천장이 높은 경우는 별로 없다. 그렇기 때문에 FOV(Field Of View)가 큰 렌즈를 사용했음에도 불구하고 사람이 비교적 크게 보이는 경우가 많다. 사람을 추적해야 하지만 사람이 크게 나타나기 때문에 사람의 전체를 바운딩 박스로 설정하는 것은 무리가 있다. 또한 사람의 위치에 따라 얼굴, 다리와 같은 부분이 안 보이는 경우가 빈번하다. 따라서 머리, 팔, 다리 등은 제외하고 몸통으로부터만 바운딩 박스를 설정하였다.

첫 번째로 추정된 바운딩 박스의 네 모서리와 실측 정보의 네 모서리 사이의 거리를 측정하는 코너 거리 오차(Euclidean distance of corners)를 계산하였다. 이는 다음 식 (8)을 이용하여 측정한다.

$$d_E = \frac{1}{4}\sum_{i=1}^{4}\sqrt{\left(u_i^M - u_i^{GT}\right)^2 + \left(v_i^M - v_i^{GT}\right)^2} \tag{8}$$

여기에서 $\left(u_i^M, v_i^M\right)$와 $\left(u_i^{GT}, v_i^{GT}\right)$, $i = 1,...,4$는 각각 예측된 사각 바운딩 박스와 실측 정보 바운딩 박스의 네 모서리를 나타낸다.

그림 8은 코너 거리 오차를 측정한 결과를 그래프로 보여주고 있다. 제안하는 방법은 노란색으로 나타내었다. Y축은 픽셀 단위로 나타내었다. 오차 결과에서는 10개의 실험 데이터에 대해 모두 200 픽셀 이하의 결과를 보여주었다. 대부분의 알고리즘이 실험 영상의 후반부로 갈수록 점차

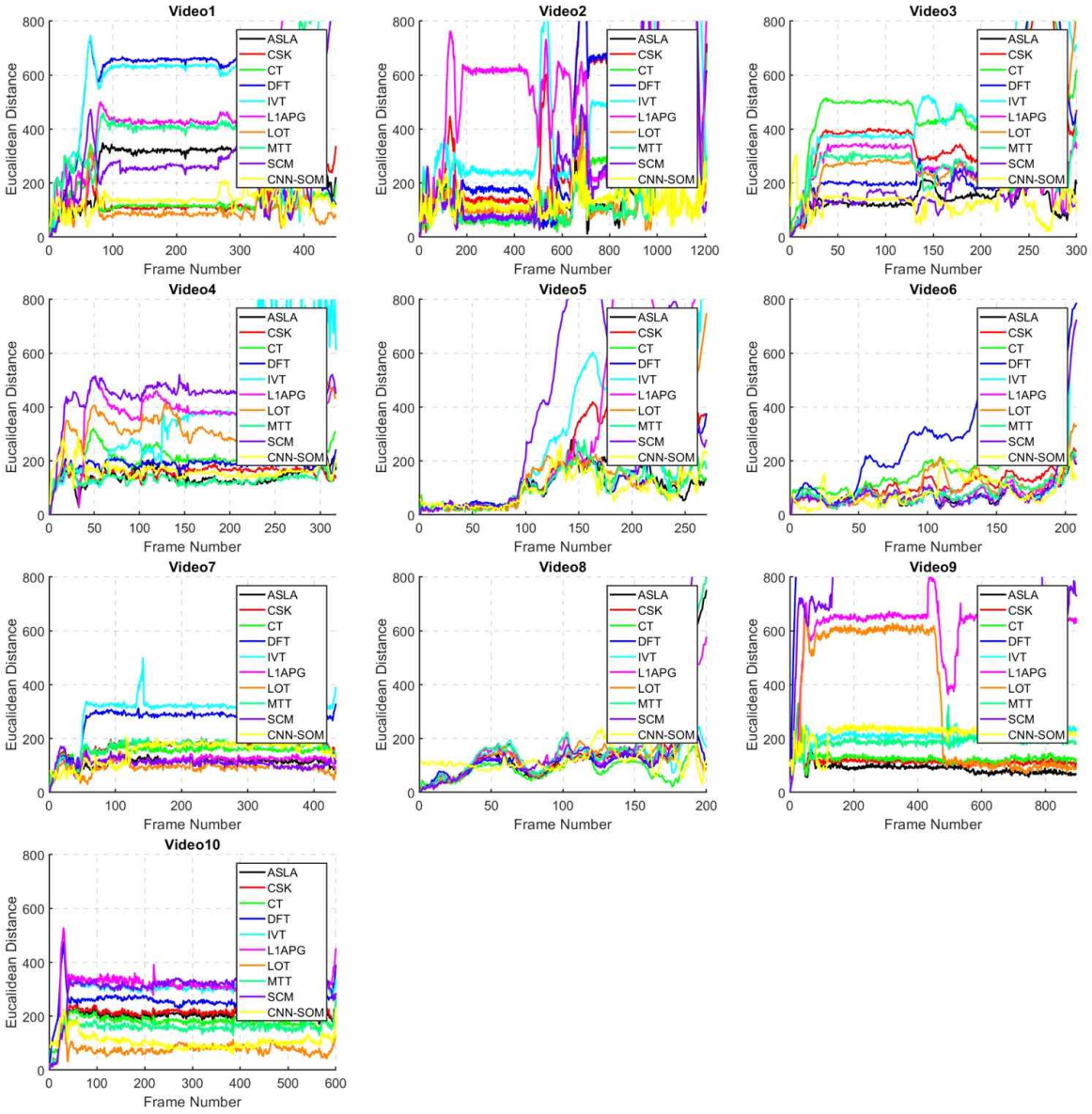

그림 8. 9개의 추적기에 대한 코너 거리 오차 측정 결과. 제안 방법은 CNN-SOM임
Fig. 8. Corner distance of the 9 trackers over 10 test sequence. CNN-SOM is the proposed method.

상승하는 결과를 보여주거나, 아니면 대부분의 프레임에서 동일한 거리를 보여주고 있다. 일부 알고리즘들은 400 픽셀이 넘는 오차를 보여주고 있다. 실험 영상은 모두 1024x786의 해상도를 가지고 있기 때문에 직접적인 비교가 가능하였다. 그러므로 유클리드 거리가 400 픽셀이 넘는 차이를 보여주는 추적기는 사실상 표류(drift)가 발생하였다고 판단할 수 있다. 또한 중간에 500 픽셀이 넘는 차이를 보였다가 후반으로 갈수록 좋은 결과를 보여주는 경우도 존재하지만, 그것은 이미 추적에 실패하였으나 중간에 우연히 올바른 추적에 복귀하였다고 볼 수 있으므로 추적에 실패했다고 판단하는 것이 옳을 것이다. 표 1에서는 각 실험 영상에 대하여 거리 오차의 평균을 구하여 나타내었다.

표 1. 코너 거리 오차의 평균
Table 1. Average of corner distance error

	ASLA	CSK	CT	DFT	IVT	L1APG	LOT	MTT	SCM	CNN-SOM
Video1	289.40	158.58	131.74	480.11	536.68	531.22	100.10	444.70	263.21	132.88
Video2	377.09	477.00	382.45	454.95	527.95	631,83	351.01	326.88	408.41	133.64
Video3	138.14	388.04	502.93	327.63	542.45	252.52	278.65	247.08	172.48	132.18
Video4	141.14	156.50	212.85	185.97	400.27	371.70	328.44	131.41	447.97	164.78
Video5	116.37	234.07	115.27	176.70	258.83	343.96	136.51	104.38	371.35	94.92
Video6	71.96	108.42	153.86	313.99	88.15	78.41	100.59	84.03	90.48	88.32
Video7	112.52	170.91	153.20	266.55	295.71	118.03	91.24	167.81	110.06	151.73
Video8	146.64	157.94	84.10	113.21	127.44	152.58	123.06	166.07	177.78	123.03
Video9	88.37	112.89	126.86	915.70	205.54	618.17	341.45	186.01	881.52	218.84
Video10	196.11	206.55	178.12	259.10	290.87	313.76	77.04	156.12	302.11	106.05

두 번째 정량적 성능 평가 도구는 중첩비(overlap ratio)이다. 중첩비의 결과는 예측된 바운딩 박스 B^M과 실측 정보 B^{GT} 사이의 중첩된 비율로 정의된다. 이는 다음 식 (9)로 표현된다.

$$R^{OR} = \frac{area(B^M \cap B^{GT})}{area(B^M \cup B^{GT})} \tag{9}$$

중첩비를 측정한 결과를 그림 9에서 보여주고 있다. 중첩비의 결과에서는 값이 높을수록 좋은 성능을 가지고 있다고 볼 수 있다. 1.0에 가까울수록 많이 중첩이 되었다는 뜻이고, 0.0에 가까울수록 본래 객체의 위치로부터 벗어남을 의미한다. 일반적으로 0.5를 임계치로 하여, 임계치보다 크면 추적에 성공, 작으면 추적에 실패하였다고 판단한다. 그러나 그래프를 보면 대부분의 경우, 최대 0.8 정도로 낮은 중첩비를 보임을 확인할 수 있다. 제안하는 방법은 비록 다른 알고리즘에 비해 높거나 유사한 결과를 보이고 있지만, 마찬가지로 약 0.5 정도의 값을 가지는 경우가 많음을 알 수 있다. 그 이유는 실험 영상의 특수성에 있다. 실험을 위해 카메라를 설치하였는데 비록 FOV가 큰 렌즈를 사용하였음에도 불구하고 대부분의 장소는 천장이 낮기 때문에 추적하고자 하는 사람의 크기가 크게 나타난다. 사람이 크기 나타나므로 사람 몸의 전체를 경계 상자로 씌우는 것은 무리가 있다. 따라서 머리, 팔 다리 등은 제외하고 몸통으로부터만 실측 정보를 생성하였다. 그러나 제안하는 추적기뿐만 아니라 대부분의 추적기들은 추적을 수행하면서 약간

표 2. 중첩비의 평균
Table 2. Average of overlap ratio

	ASLA	CSK	CT	DFT	IVT	L1APG	LOT	MTT	SCM	CNN-SOM
Video1	0.1818	0.3728	0.4087	0.1062	0.0821	0.0682	0.4962	0.0730	0.1818	0.4183
Video2	0.3842	0.1834	0.3356	0.2207	0.1095	0.0584	0.3490	0.4048	0.2792	0.4214
Video3	0.4617	0.0967	0.0480	0.2887	0.0922	0.2564	0.2979	0.2773	0.3796	0.4235
Video4	0.4259	0.3982	0.2679	0.3329	0.1332	0.0642	0.1952	0.4464	0.0399	0.3616
Video5	0.5244	0.3732	0.5314	0.4703	0.3635	0.3834	0.5151	0.5591	0.3097	0.5963
Video6	0.6883	0.5716	0.4439	0.2526	0.6666	0.6715	0.5873	0.6402	0.6682	0.6261
Video7	0.4149	0.2913	0.3732	0.1511	0.0740	0.4116	0.4433	0.3204	0.4249	0.3248
Video8	0.4381	0.3989	0.5610	0.4604	0.3758	0.3928	0.3963	0.3874	0.3894	0.4492
Video9	0.5358	0.4799	0.4282	0.0039	0.1935	0.0163	0.2341	0.2653	0.0087	0.2452
Video10	0.2865	0.2699	0.3313	0.1759	0.1364	0.0572	0.5427	0.4164	0.0889	0.5369

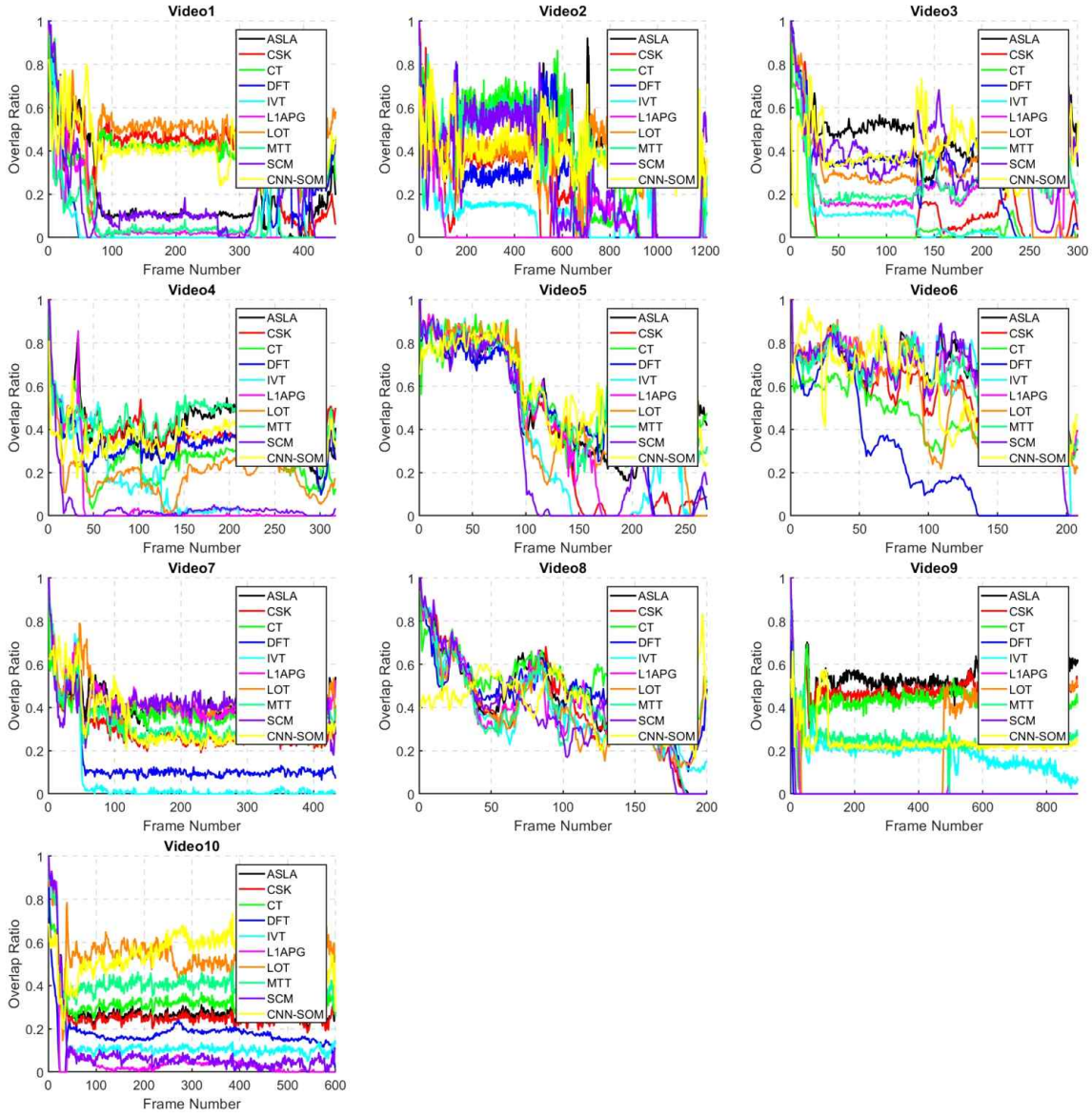

그림 9. 9개의 추적기에 대한 중첩 비 측정 결과
Fig. 9. Overlap ratio of the 9 trackers over 10 test sequence.

씩의 오차가 누적이 되면서 몸통만을 추적하는 것이 아니라 사람 전체를 추적하게 된다. 따라서 중첩 비는 기존의 실험에 사용되었던 영상들과 비교하여 낮은 값을 가질 수 밖에 없다. 그럼에도 불구하고 제안방법은 다른 방법들과 비교하였을 때, 우수한 결과를 보여주고 있다. 표 2에서는 각 실험 영상에 대하여 중첩비의 평균을 구하여 나타내었다.

V. 결 론

본 논문에서는 합성곱 신경망과 자기 조직화 지도를 활용하여 점유 센서 영상에서 사람을 추적하는 기법을 제안하였다. 기존의 온라인 객체 추적에서는 많은 수의 훈련 집합을 필요로 하지만 제안 기법에서는 적은수의 훈련 집합으로도 충분히 학습 및 추적이 가능함을 증명하였다. 또한

컬러 정보를 결합하여 새로운 손실함수를 정의함으로써, CNN 모델 학습의 효율을 향상시켰다. 제안하는 CNN 기반 객체 추적을 넘어서 성능을 향상시키기 위해 변형된 SOM 알고리즘을 적용하였다. SOM의 적용은 후보 이미지 패치들을 선별함으로써 연산량을 줄일 수 있었을 뿐만 아니라 추적기의 성능 또한 개선시킬 수 있다. 실험 영상에 대하여 제안하는 추적기를 적용하였고, 정량적 평가를 위해 코너 거리 오차와 중첩비를 측정하였다. 코너 거리 오차에서 다른 알고리즘과 비교한 결과, 총 10개의 실험 영상에 대해 각각 3, 1, 1, 4, 1, 5, 5, 3, 5, 2의 순위를 기록하였고, 총 3개의 비디오에서 가장 좋은 결과를 보여주었다. 또한 중첩비에 대해서도 동일한 방법으로 평가한 결과 2, 1, 2, 4, 1, 6, 6, 3, 5, 2의 순위를 기록하였고, 총 2개의 비디오에서 가장 좋은 중첩비를 확인할 수 있었다. 향후 제안하는 추적 기법을 이용하여 감시 시스템과 같은 분야에서 유용하게 활용될 수 있을 것으로 기대된다.

참 고 문 헌 (References)

[1] P. Liu, S. Nguang, and A. Partridge, “Occupancy inference using pyro-electric infrared sensors through hidden markov model“, *IEEE Sensors Journal*, 16(4), Feb, 2016.

[2] F. Wahl, M. Milenkovic, and O. Amft, “A distributed PIR-based approach for estimating people count in office environments“, *IEEE Conf. on Computational Science and Engineering*, 2012.

[3] Y. Benezeth, H. Laurent, B. Emile, and C. Rosenberger, “Towards a sensor for detecting human presence and characterizing activity“, *Energy and Buildings*, 43, 2011.

[4] J. Han, and B. Bhanu, “Fusion of color and infrared video for moving human detection“, *Pattern Recognition*, 40, 2007.

[5] S. Nakashima, Y. Kltazono, L. Zhang, and S. Serikawa, “Development of privacy-preserving sensor for person detection“, *Proceedia-Social and Behavioral Sciences*, 2(1)n, 2010.

[6] I. Amin, A. Taylor, F. Junejo, A. Al-Habaibeh, and R. Parkin, “Automated people-counting by using low-resolution infrared and visual cameras“, *Measurement*, 41, 2008.

[7] J. Gil, and M. Kim, “Real-time People Occupancy Detection by Camera Vision Sensor“, *Journal of Broadcast Engineering*, 22(6), pp. 774-784, 2017.

[8] H. Li, Y. Li, and F. Porikli, “DeepTrack: Learning Discriminative Feature Representations Online for Robust Visual Tracking“, *IEEE Trans. on Image Processing*, Vol. 25, No. 4, pp. 1834-1848, April 2016.

[9] K. Zhang, Q. Liu, and M. Yang, “Robust Visual Tracking via Convolutional Networks Without Training”, *IEEE Trans. on Image Processing*, Vol. 25, No. 4, pp. 1779-1792, April 2016.

[10] X. Zhou, L. Xie, P. Zhang, and Y. Zhang, “An Ensemble of Deep Neural Networks for Object Tracking”, *IEEE Conf. on Image Processing*, pp. 843-847, 2014.

[11] T. Kohonen, “Self-organized formation of topologically correct feature maps“, Biological cybernetics, 43(1), pp. 59-69, 1982.

[12] X. Jia, H. Lu and M. H. Yang, “Visual Tracking via Adaptive Structural Local Sparse Appearance Model“, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1822-1829, 2012.

[13] J. F. Henriques, R. Caseiro, P. Martin and J. Batista, “Exploiting the Circulant Structure of Tracking-by-Detection with Kernels“, *European Conf. on Computer Vision*, pp. 702-715, 2012.

[14] K. Zhang, L. Zhang and M. H. Yang, “Real-time Compressive Tracking“, *European Conf. on Computer Vision*, pp. 864-877, 2012.

[15] L. Sevilla-Lara and E. Learned-Miller, “Distribution Fields for Tracking“, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1910-1917, 2012.

[16] D. A. Ross, J. Lim, R. S. Lin and M. H. Yang, “Incremental Learning for Robust Visual Tracking“, *International Journal of Computer Vision*, Vol. 77, Issue 1-3, pp. 125-141, 2008.

[17] C. Bao, Y. Wu, H. Ling and H. Ji, “Real Time Robust L1 Tracking Using Accelerated Proximal Gradient Approach“, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1830-1837, 2012.

[18] S. Oron, A. Bar-Hillel, D. Levi and S. Avidan, “Locally Orderless Tracking“, *International Journal of Computer Vision*, Vol. 111, No. 2, pp. 213-228, 2015.

[19] T. Zhang, B. Ghanem, S. Liu and N. Ahuja, “Robust Visual Tracking via Multi-task Sparse Learning“, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2042-2049, 2012.

[20] W. Zhong, H. Lu and MH. Yang, “Robust Object Tracking via Sparsity-based Collaborative Model“, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1838-1845, 2012.

저 자 소 개

길 종 인

- 2010년 8월 : 강원대학교 컴퓨터정보통신공학과 학사
- 2012년 8월 : 강원대학교 컴퓨터정보통신공학과 석사
- 2012년 9월 ~ 현재 : 강원대학교 IT대학 컴퓨터정보통신공학과 박사과정
- 주관심분야 : 객체 트랙킹, 딥러닝, 점유센서, 머신러닝

김 만 배

- 1983년 : 한양대학교 전자공학과 학사
- 1986년 : University of Washington, Seattle 전기공학과 공학석사
- 1992년 : University of Washington, Seattle 전기공학과 공학박사
- 1992년 ~ 1998년 : 삼성종합기술원 수석연구원
- 1998년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 교수
- 2016년 ~ 현재 : 강원대학교 정보통신연구소 소장
- ORCID : http://orcid.org/0000-0002-4702-8276
- 주관심분야 : 3D영상처리, 비전점유센서, 객체인식