

A Method of Tracking Object using Particle Filter and Adaptive Observation Model

Hyoyeon Kim *, Kisang Kim **, Hyung-Il Choi***

Abstract

In this paper, we propose an efficient method that is tracking an object in real time using particle filter and adaptive observation model. When tracking object, it happens object shape variation by camera or object movement in variety environments. The traditional method has an error of tracking from these variation, because it has fixed observation model about the selected object by the user in the initial frame. In order to overcome these problems, we propose a method that updates the observation model by calculating the similarity between the used observation model and the eight-way of edge model from the current position. If the similarity is higher than the threshold value, tracking the object using updated observation model to reset observation model. On the contrary to this, the algorithm which consists of a process is to maintain the used observation model. Finally, this paper demonstrates the performance of the stable tracking through comparison with the traditional method by using a number of experimental data.

▶ Keyword : Particle Filter, observation model, update, object tracking

I. Introduction

객체 추적에 대한 연구는 활발히 진행되어 있고, 이 기술은 공공기관의 감시 시스템, 증강현실, 행동인식, 스마트 홈, 로봇추적 등과 같이 컴퓨터 비전 분야에서 중요한 역할을 한다[1]. 일반적으로 카메라의 떨림, 조명의 변화, 물체의 형태의 변화, 복잡한 배경과 같은 많은 외부적 요인이 추적 성능에 많은 영향을 받게 된다. 이러한 문제점을 해결하기 위해 다양하게 변화하는 외부 환경에 대해서 물체를 정확하게 추적할 수 있도록 많은 연구가 진행되고 있다[2]. 본 논문에서는 다양한 알고리즘 중 파티클 필터(Particle filter)를 이용하여 정확성을 높인 객체를 추적하는 방법에 대해 연구하였다. 기존 연구에서는 초기에 8방향 관측 모델을 생성한 후, 그 모델을 고정하여 객체 추적에 사용하였다[3]. 단순한 에지를 가진 객체와 에지의 변화가 적은 객체의 경우에는 추적이 가능하지만, 카메라 또는 객체의 움직임으로 인한 객체의 크기, 회전의 변화가 클 경우에는 추적의 오차가 생겨 정확성이

떨어진다. 이러한 문제점을 해결하기 위해 본 논문에서는 객체의 회전 변화에 영향을 적게 받고, 추적의 정확성을 높이기 위해 관측 모델을 적응적으로 업데이트하는 방법을 제안한다.

그림 1의 시스템 흐름도와 같이 먼저, 초기 프레임에서 추적하고자 하는 객체를 사용자가 지정한다. 지정된 영역 안에 있는 객체에 대해 8방향 에지 관측 모델을 생성한다. 초기 N개의 파티클 집합을 생성한 후, 이 파티클 집합과 초기에 생성한 에지 관측 모델을 기반으로 객체의 위치인 상태를 예측한다. 다음으로 기존 연구에서 제안되었던 업데이트 단계는 현재 추적에 사용되고 있는 기존 관측 모델과 각 파티클의 8방향 에지 모델의 거리를 비교하여 해당 파티클의 가중치를 업데이트한다. 또한, 리샘플링 단계에서는 업데이트된 가중치를 이용하여 파티클을 리샘플링하고, 객체의 상태를 갱신하여 현재 위치를 추정한다. 마지막으로, 본 논문에서 제안하는 방법인 관측 모델 업데이트 단계는 현재 추적에 사용되는 기존 관측 모델을 재설정할지 결정하고, 객체 추적을

• First Author: Hyoyeon Kim, Corresponding Author: Hyung-Il Choi
*Hyoyeon Kim(khy_15@ssu.ac.kr), Dept of Global Media, Soongsil University.
**Kisang Kim(kimkisang@ssu.ac.kr), Dept of Global Media, Soongsil University.
***Hyung-Il Choi(hic@ssu.ac.kr), Dept of Global Media, Soongsil University.
• Received: 2016. 10. 12, Revised: 2016. 11. 21, Accepted: 2016. 12. 21.

진행하게 된다. 이 단계에서는 추정된 위치를 중심으로 기존 관측 모델과 동일한 크기의 영역에 대해 Grassfire 알고리즘을 이용하여 영역 안의 객체를 분할한다. 분할된 객체는 Central Moment를 이용해 객체의 회전각으로 회전된 후, 기존 관측 모델과 추정된 위치에서의 8방향 모델과의 유사도에 대해 계산된다. 두 모델에 대한 유사도가 임계값보다 클 경우, 관측 모델은 재설정되고 업데이트된 관측 모델을 이용하여 객체를 추적이 진행된다. 반대로, 유사도가 임계값보다 작을 경우, 기존 관측 모델을 유지하여 알고리즘이 진행되는 과정으로 이루어진다.

위와 같은 내용으로 본 논문의 구성은 다음과 같다. 2장에서는 제안하는 관측 모델 업데이트 단계로, 파티클 필터의 관측 모델을 적응적으로 업데이트 하는 방법에 대해 설명한다. 3장에서는 실험 결과를 통해 고정된 관측 모델을 가진 파티클 필터와 제안하는 적응적으로 관측 모델을 업데이트하는 방법을 가진 파티클 필터 간의 성능을 입증하고, 4장에서는 본 논문에 대한 결론에 대해 기술하였다.

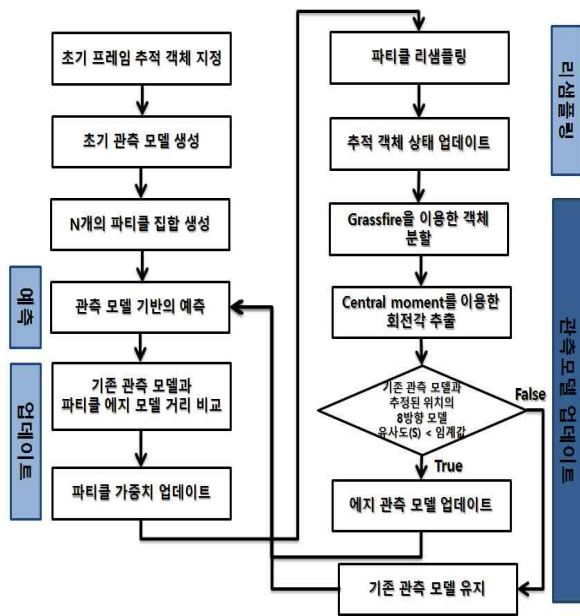


Fig. 1. 시스템 전체 흐름도

II. A method of updating adaptive observation model in particle filter

파티클 필터 알고리즘의 관측 모델은 파티클의 가중치를 측정하는데 사용되고, 객체 추적을 위해 중요한 요소이다. 많은 관측 모델들은 파티클 필터링 추적을 위해 만들어진다. 또한, 관측 모델은 다양한 특징을 이용해 생성될 수 있으며, 본 논문에서는 영상의 색상과 조명 변화에 대한 영향이 적은 에지를 이용하여 관측 모델을 생성하였다.

8방향 에지 관측 모델을 사용하여 객체를 추적하는 기존 연구에서는 초기 프레임에서 8방향 에지 관측 모델을 생성한 후, 그 모델을 고정하여 추적에 사용한다[3]. 그러나 고정된 관측 모델을 사용

하여 객체를 추적할 경우에는 객체의 크기, 회전 등의 변화에 영향을 많이 받아서 추적의 오차가 생긴다. 본 논문에서는 고정된 관측 모델로 인한 추적의 오차를 줄이기 위해 적응적으로 관측 모델을 업데이트하는 방법을 제안한다. 이 관측 모델 업데이트 단계는 추적 객체 상태인 위치가 갱신 된 후, 기존 관측 모델과 추정된 위치에서의 8방향 모델과의 유사도를 계산하여 관측 모델을 업데이트 유무를 설정하기 위해 사용된다. 현재 추적에 사용되는 관측 모델인 기존 관측 모델과 현재 추정된 위치에서의 8방향 모델과의 유사도를 계산하여 이 유사도가 임계값 이상인 경우에는 관측 모델을 업데이트한다. 업데이트된 관측 모델은 파티클 필터 알고리즘의 업데이트 단계에서 각 파티클의 가중치를 업데이트하는데 사용된다. 반대로, 이 유사도가 임계값 이하인 경우에는 기존 관측 모델과 현재 위치에서의 8방향 모델이 유사함을 의미하기 때문에 기존에 사용되고 있는 관측 모델을 유지하여 객체를 추적한다.

1. Grassfire를 이용한 추적 객체 분할

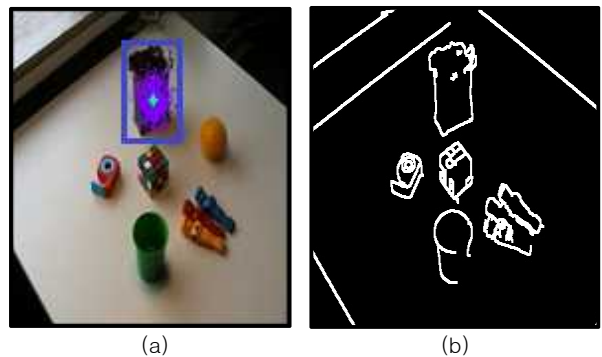


Fig. 2. 추정된 위치에서의 객체 영역 (a) 현재 프레임 원본 이미지 (b) (a)의 에지를 추출한 이미지

추적하고자 하는 객체의 회전에 의한 변화를 알기 위해 먼저, 현재 프레임 이미지인 Fig 2의 (a)에 대한 에지를 추출해 (b)와 같은 이미지를 만든다. 본 논문에서는 에지를 추출하기 위해 Canny Operation을 사용하였다. (a)에서 초록색 십자 모양으로 표시된 현재 추정된 위치를 중심으로 한 기존 관측 모델과 크기가 같은 영역을 지정한다. 이 지정된 영역은 Fig 2의 (a)에 파랑색 네모 박스로 표시되었다. 객체는 현재 추정된 위치인 초록색 십자 중심에서 시작하여 객체 에지까지 Grassfire 알고리즘을 이용하여 분할될 수 있다. 분할된 객체는 Fig 3의 회색 부분으로 표현된 것을 확인할 수 있다.



Fig. 3. Grassfire를 이용한 객체 분할

Table 1. Grassfire 알고리즘

Procedure Grassfire
Input : imagedata data; integer cx, cy, width, height;
If(cx < 0 cy < 0 cx >= width cy >= height) return;
data(cx, cy) = 0;
If(data(cx+1, y) is 1) Grassfire(data, width, height, cx, cy);
If(data(cx-1, y) is 1) Grassfire(data, width, height, cx, cy);
If(data(cx, y+1) is 1) Grassfire(data, width, height, cx, cy);
If(data(cx, y-1) is 1) Grassfire(data, width, height, cx, cy);

표 1은 객체를 분할하기 위해 사용되는 Grassfire 알고리즘이다. Input값으로 data는 Fig 2의 (a)에서 객체를 포함한 파란색 네모 박스 영역의 에지를 추출한 Fig 2의 (b)인 박스 영역이 포함된 에지 이미지이고, cx, cy는 Fig 2의 (a)에서 초록색 십자 모양인 현재 추정된 위치를 의미한다. 또한, width, height는 (a)에서 파란색 네모 박스로 표현된 data의 width, height의 크기를 의미한다. Grassfire 알고리즘은 현재 추정된 위치인 cx, cy을 시작점으로 data의 범위만큼 수행한다. 이것은 해당 픽셀의 4방향으로 탐색하여 픽셀 값이 1이면 Grassfire함수가 재귀되는 과정으로 객체의 에지 안쪽 부분을 분할할 수 있다. 이 알고리즘을 이용하면 Fig 3과 같이 회색 부분으로 표현된 분할된 객체를 만들 수 있다. 분할된 객체는 Central Moment를 이용해 회전각을 추출하는데 사용된다.

2. Central Moment를 이용한 회전각 추출

Grassfire 알고리즘을 통해 분할된 객체와 Central Moment를 이용해 객체의 회전된 각도를 추출할 수 있다. 먼저, 분할된 이미지를 통해 Moment 값을 구할 수 있다. 식 1은 이미지의 위치를 이용해 Moment 값을 구하는 수식이다. $I(x,y)$ 은 그레이스케일 이미지에서의 명암 값을 의미한다. 본 논문에서 $I(x,y)$ 는 Grassfire로 분할된 이진화된 영상을 사용한다.

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \quad [i, j \in 0, 1, 2, \dots] \quad (1)$$

객체의 회전각을 구하기 위해서는 영상의 (p, q) 차원 Central Moment인 μ_{pq} 을 이용한다. 이것은 객체 영역의 중심, 크기와 방향을 구하기 위해 사용되는 영역의 모멘트(moments) 계산에서 유도된 식이다[4]. 또한, Moment invariants는 Central moment의 함수들이기 때문에 모양 변화에 불변하기 위해 사용된다.[5]. 본 논문에서는 회전에 불변하다는 특성을 이용하기 위해 Central moment를 사용하였다. 식 3에서 필요한 μ_{20} , μ_{02} , μ_{11} 를 구하기 위해 식 2를 이용하였다. μ_{20} 은 x축으로부터의 거리에 대한 분산이고, μ_{02} 은 y축으로부터의 거리에

에 대한 분산을 나타낸다. 또한, μ_{11} 은 x좌표와 y좌표에 대한 상관관계수인 공분산을 나타낸다. 식 2에서 객체의 수평 및 수직 방향의 무게 중심은 각각 \bar{x} , \bar{y} 으로 표현된다.

$$\begin{aligned} \mu_{20} &= \left(\frac{M_{20}}{M_{00}} \right) - (\bar{x})^2 \\ \mu_{02} &= \left(\frac{M_{02}}{M_{00}} \right) - (\bar{y})^2 \\ \mu_{11} &= \left(\frac{M_{11}}{M_{00}} \right) - (\bar{x} * \bar{y}) \end{aligned} \quad (2)$$

그 다음으로 식 2를 통해 구한 μ_{20} , μ_{02} , μ_{11} 은 식 3에 대입해 회전각을 구할 수 있다.

$$\Theta = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \quad (3)$$

Fig 4의 (a), (b)에 표시된 초록색 직선은 식 3를 통해 구한 회전각 Θ 과 무게중심 \bar{x} , \bar{y} 을 이용하여 구할 수 있다. 이 직선은 무게중심을 통과하는 영역의 기울기를 나타내는 직선이다. Fig 5의 (a), (b)와 같이 회전각이 보정된 객체는 이 직선을 빨간색으로 표현한 y축으로 회전하여 만들 수 있다.

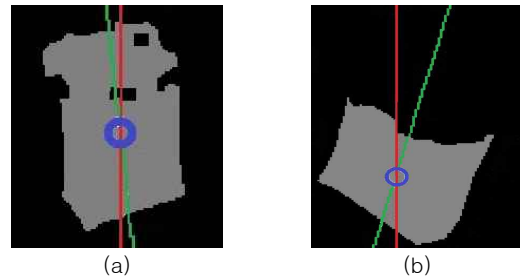


Fig. 4. Central Moment를 이용한 회전각 추출
(a) : 기존 관측 모델, (b) : Frame 550

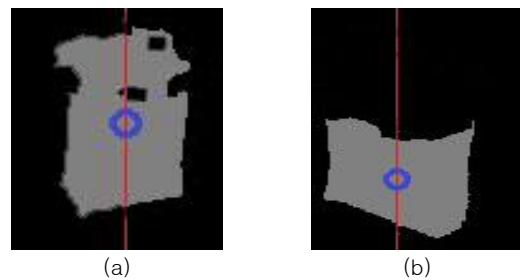


Fig. 5. 추출된 회전각으로 회전된 객체
(a) : 기존 관측 모델, (b) : Frame 550

3. 기존 관측 모델과 현재 추정된 위치에서의 8방향 에지 모델의 유사도 비교

기존 관측 모델은 현재 파티클 필터의 추적이 사용되고 있는 관측 모델을 의미한다. 관측 모델 M 는 Fig 6과 같이 사용자가 지정한 영역의 중심으로부터 45°씩 8방향으로 가장 가까운 에지를 탐색하여 8방향 에지 모델로 만들어진다. 이 8방향 에

지 모델은 객체를 추적하기 위해 사용되는 관측 모델 (Observation model)이다. 관측 모델 M 는 식 4와 같이 각 방향에 대한 각도 θ_j 와 중심에서 가장 가까운 에지까지의 거리인 $mDis_j$ 를 포함한다[3].



Fig. 6. 회전이 보정된 객체의 기존 관측 모델

$$M = \{mDis_j, \theta_j\} \quad [j \in 1..8] \quad (4)$$

Fig 7은 Central Moment를 통해 회전이 보정된 8방향 에지 모델 U 이다. 이 8방향 모델 U 은 기존 관측 모델과의 유사도를 비교하기 위해 사용된다. 본 논문에서 사용되는 현재 추정된 위치를 중심으로 생성된 8방향 모델 U 은 관측 모델 M 과 같이 각 방향에 대한 각도 θ_j 와 중심에서 가장 가까운 에지까지의 거리인 $uDis_j$ 를 포함하여 식 5와 같이 표현된다.

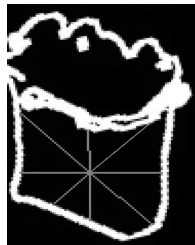


Fig. 7. 회전이 보정된 객체의 8방향 에지 모델

$$U = \{uDis_j, \theta_j\} \quad [j \in 1..8] \quad (5)$$

본 논문에서는 기존 관측 모델 M 과 현재 추정된 위치의 8방향 모델 U 을 비교한 유사도 D 를 계산한다. 유사도 D 는 식 6과 같이 기존 관측 모델의 거리 $mDis_j$ 와 현재 추정된 위치의 8방향 모델의 거리 $uDis_j$ 의 차를 8로 나누고, 제곱하여 각 방향에 따른 거리의 차이를 계산하여 구할 수 있다. 식 6의 M 과 U 의 각 각도에 따른 거리는 SIFT 알고리즘(Scale Invariant Feature Transform)의 Key matching하는 부분에서 각 키 간의 거리를 구하는 식과 유사한 방법으로 구성하였다. SIFT 알고리즘은 영상의 크기와 회전에 불변하는 특징점을 추출하는 알고리즘이다. 이것은 본 논문에서의 8방향 에지 모델이 SIFT의 서술자를 생성할 때 8개의 각도로 양자화하여 특징점에 대한 특징벡터로 재구성하는 것[7]과 모양이 유사하고, SIFT는 회전에 불변한 특징을 가지고 있기 때문에 유사도 D 구하는 식 6은 SIFT와 유사한 방법으로 구성하였다. SIFT에서 Image gradient magnitude의 값은 유클리드 거리(Euclidean Distance)를 이용하여 구하는데[8], 본 논문에서는 유사도 D 를

구하기 위해 8로 나눈 값의 루트 값을 사용하였다. 유사도 D 의 값은 작을수록 기존 관측 모델 M 과 유사함을 의미한다.

$$D = \sqrt{\frac{\sum_{i=0}^7 (mDis_i - uDis_i)^2}{8}} \quad (6)$$

유사도 D 가 임계값보다 작을 경우 기존에 사용되고 있는 관측 모델을 유지하여 파티클 필터 알고리즘은 수행된다. 반대로, 유사도 D 가 임계값보다 클 경우, 관측 모델을 재설정하여 업데이트된 관측 모델은 다음 프레임부터 객체의 위치를 예측하는 단계에서 사용되어 진다. 본 논문에서는 추적하고자 하는 객체에 따라 사용자가 지정한 임계값을 사용하였다. 이 임계값은 추적하고자 하는 객체에 대해 사용자가 지정한 값이다. 본 실험영상에서는 임계값을 약 5.8로 지정하였다. 관측 모델 업데이트 할 때 관측 모델은 사용자가 마우스로 지정하여 재설정된다. 관측 모델 재설정 후, 이 업데이트된 관측 모델을 이용하여 추적 객체는 이전 위치를 중심으로 한 10X10 주변에 파티클을 랜덤하게 뿌려서 찾게 된다. 동영상에서 추적하는 객체의 위치 변화는 크지 않기 때문에 이전 위치의 10X10 주변을 탐색하여 파티클 필터 알고리즘이 수행된다.

Table 2. 회전 변화에 따른 유사도 비교

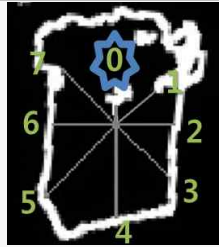
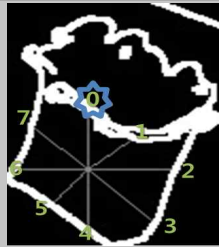
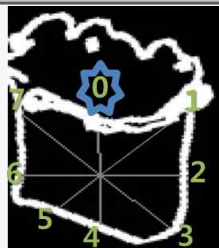
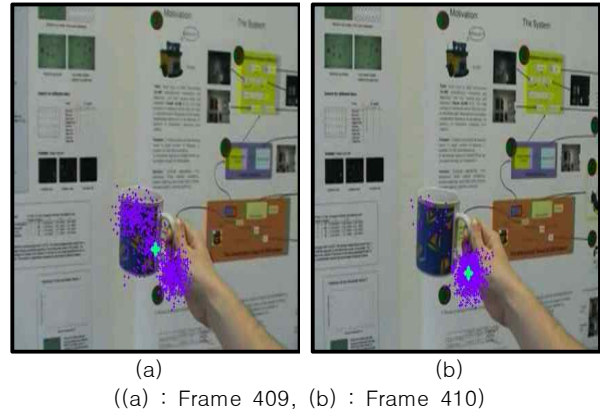
	0	23.0
	1	61.8
	2	71.0
	3	86.3
	4	110.0
	5	118.7
	6	79.0
	7	96.4
(a) 기존 관측 모델		
	0	30.0
	1	26.2
	2	39.0
	3	42.4
	4	33.0
	5	24.8
	6	32.0
	7	34.6
(b) 회전 보정되지 않은 객체 (Frame 550)		
	0	56.0
	1	105.6
	2	100.0
	3	110.4
	4	62.0
	5	66.0
	6	95.0
	7	123.8
(c) 회전 보정된 객체 (Frame 550)		

표 2는 회전이 보정되지 않은 객체의 8방향 모델과 제안하는 방법을 적용해 회전이 보정된 객체의 8방향 모델을 기존 관측 모델과 유사도를 비교한 표이다. (a)는 추적에 사용되고 있는 기존 관측 모델이고, 0을 기준으로 8방향 길이 값은 0부터 7까지 표현하였다. 또한, 회전이 보정되지 않은 객체인 (b)와 회전이 보정된 객체인 (c)은 0을 기준으로 한 8방향 길이 값을 각 그림 옆의 표로 표현하였다. (a)와 (b), (a)와 (c) 두 모델 간의 유사도를 비교하기 위해 본 논문에서는 제안하는 식 6을 통해 유사도를 구했다. 기존 관측 모델(a)와 회전이 보정되지 않은 객체의 8방향 모델(b)와의 유사도는 55.9이고, 회전이 보정된 객체의 8방향 모델(c)와의 유사도는 36.2이다. (a)-(c)의 유사도는 (a)-(b)의 유사도보다 작은 값으로 (a)-(c)의 모델이 더 유사함을 확인할 수 있다. 따라서, 표2의 결과는 제안하는 방법으로 객체의 회전 변화를 보정함으로써 추적하고 있는 객체의 회전 변화에 따른 추적의 영향을 적게 받음을 보인다.

가 생기게 되면 초기에 생성한 관측 모델과 차이가 생기게 때문에 추적의 오차가 점차 커지게 된다. 그러나 그림 8의 (c), (d)에서 보듯이 본 논문에서 제안하는 적응적으로 관측 모델을 업데이트하는 방법을 가진 파티클 필터는 객체의 여러 변화에 따른 차이를 관측 모델을 업데이트함으로써 추적의 오차를 개선하여 객체 추적의 정확성을 높인 결과를 보여주고 있다.



III. Experiment

본 논문에서는 관측 모델 업데이트 단계를 포함한 파티클 필터의 성능을 평가하기 위해 Klein[7]의 실험에서 사용한 5개의 동영상을 본 실험에 사용하였다. 각 동영상은 움직이는 카메라 환경에서 촬영되었으며 객체의 형태도 연속적으로 변화하는 동영상을 포함하고 있다. 실험을 위해 본 논문에서는 고정된 에지 관측 모델을 가진 파티클 필터와 적응적으로 관측 모델을 업데이트하는 방법을 가진 파티클 필터의 성능을 비교하였다.

그림 8은 두 개의 파티클 필터를 이용해 객체를 추적한 결과이다. 이 실험에서는 움직이는 카메라와 객체의 환경을 가진 Video_1을 사용하였다. 고정된 에지 관측 모델을 가진 파티클 필터를 이용해 추적하게 되면 객체가 가진 여러 무늬로 인한 색 혼합도, 객체와 유사한 배경의 의한 추적의 혼돈은 적다. 그러나 이 파티클 필터로 추적한 결과인 (a), (b)와 같이 카메라 또는 객체의 움직임으로 인해 객체의 크기, 회전과 모양의 변화

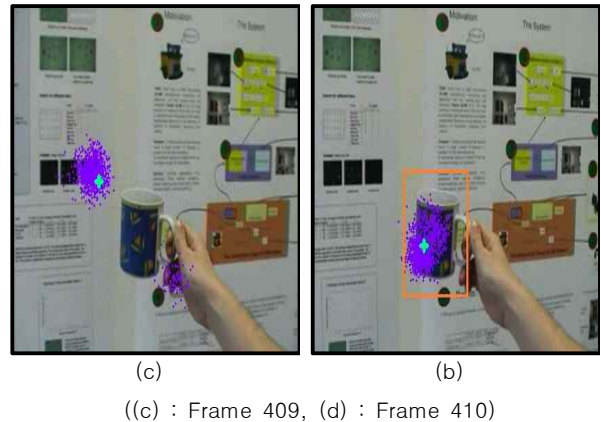


Fig. 8. 객체 추적 결과 (a),(b) 고정적 에지 관측 모델을 가진 파티클 필터 (c),(d) 적응적 에지 관측 모델을 가진 파티클 필터

Table 3. The average performance comparison using 5 test videos

Name	Total Frame	고정된 에지 관측 모델 [fixed]		적응적 에지 관측 모델 [Proposed]	
		Correct	%	Correct	%
Video_1	645	538	83.4	596	92.4
Video_2	686	537	78.2	602	87.7
Video_3	824	692	83.9	741	89.9
Video_4	396	299	75.5	334	84.3
Video_5	700	547	78.1	623	89.0
Avg			79.8		88.6

그림 9과 표 3에서는 (1) 고정된 관측 모델을 가진 파티클 필터를 fixed로 표기하였고, (2) 적응적으로 관측 모델을 업데이트하는 방법을 가진 파티클 필터를 Proposed로 표현하였다. 표 3에서 보듯이, 각 동영상에서 추적하고자 하는 객체를 제대로 추적이 된 프레임의 수는 확률로 표현하여 성공률에 대한 평균으로 나타낸다. 표 3에서 Correct는 객체가 추적에 성공한 영상 프레임의 수를 나타내고, %는 Total Frame에 대한 Correct의 확률인 성공률을 나타낸다. 또한, Avg는 5개의 동영상에 대한 성공률 %의 평균을 나타내고, 방법 (2)의 Avg는 방법 (1)의 Avg보다 약 10%정도 높아짐이 확인되어 추적에 대한 성능이 개선된 것으로 볼 수 있다.

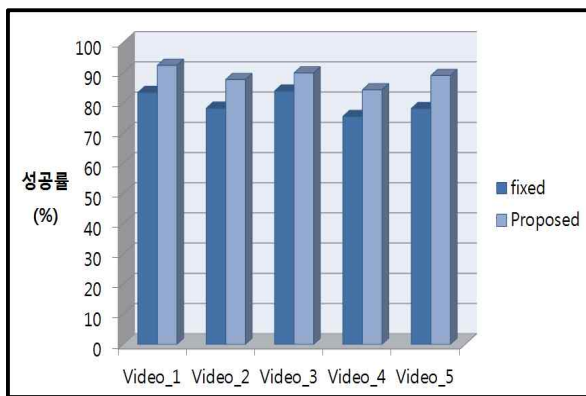


Fig. 9. 두 개의 알고리즘 성능 평가 그래프

그림 9은 표 3을 보기 쉽게 막대그래프로 표현한 것이다. (1)fixed는 진한 파랑색으로, (2)Proposed는 연한 파랑색 그래프로 표현하였다. 그림 9의 그래프의 가로축은 각 동영상을 나타내고, 세로축은 %인 총 프레임 수에 대한 추적을 성공한 프레임의 수의 확률인 성공률을 나타낸다. 방법 (1)과 방법 (2)의 성능 차이는 최소 6%에서 최대 10.1%의 성능이 나아짐을 확인할 수 있다.

성능 결과표와 그래프에서 보는 바와 같이 본 논문에서 제안한 적응적으로 관측 모델을 업데이트하는 방법을 가진 파티클 필터는 고정된 관측 모델을 가진 파티클 필터에 비해 높은 성능을 나타냄을 확인할 수 있다. 고정된 관측 모델을 가진 파티클 필터는 객체의 크기, 회전과 모양 변화에 의해 프레임의 수가 많아질수록 추적의 오차가 생기게 된다. 그러나 본 논문에서 제안하는 알고리즘을 이용하여 객체를 추적하면 객체의 여러 변화에 대한 영향을 받지 않기 위해 관측 모델을 업데이트하기 때문에 추적에 대한 오차가 개선되어 추적의 정확성이 우수함을 확인할 수 있다.

IV. Conclusion

본 논문에서는 예지로 생성한 8방향 관측 모델을 적응적으로 업데이트하는 파티클 필터에 적용하여 실시간 객체를 추적

하는 방법을 제안하였다. 이 방법은 8방향 예지 관측 모델을 이용한 파티클 필터로 객체의 위치를 추적하고, 8방향 예지 관측 모델과 추정된 위치에서 8방향 예지 모델의 유사도를 측정하여 유사도가 임계값 이상일 때 관측 모델을 업데이트한다. 업데이트된 관측 모델은 파티클 필터 알고리즘의 예측 단계에서 사용되고, 그 모델을 이용해 추적이 진행된다. 제안하는 파티클 필터는 예지 특징을 사용해 추적하는 객체의 관측 모델을 효율적으로 모델링하고, 객체의 변화에 대해 적응적으로 업데이트함으로써 그림 9과 표 3의 결과와 같이 기존의 방법보다 더 나은 추적에 대한 성능을 보여주었다. 또한, 표 2의 결과에서 보는 바 같이 본 논문에서 제안하는 방법을 적용한 회전이 보정된 객체는 회전이 보정되지 않은 객체보다 동일한 객체에 대해 유사도를 높게 계산될 수 있다. 그 결과, 적응적으로 예지 관측 모델을 업데이트하는 방법으로 추적하였을 때 객체 추적의 정확성이 개선되었다. 향후 제안한 방법으로는 여러 개의 관측 모델을 학습하여 다중객체 추적과 행동 추적, 로봇 추적 등의 다양한 분야에 활용될 수 있을 것이다.

REFERENCES

- [1] B. C. Ko, J. Y. Nam, J. Y. Kwak, "Object Tracking Using Particle Filters in Moving Camera", The Korean Institute of Communications and Information Sciences, Vol.37A, No. 05, pp.375-387, 2012.
<http://dx.doi.org/10.7840/KICS.2012.37A.5.375>
- [2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," ACM Computing Surveys, vol. 38, no. 4, Dec. 2006.
<http://doi.acm.org/10.1145/1177352.1177355>
- [3] H. Y. Kim, K. S. Kim, H. Y. Choi, "Tracking moving objects using particle filter and edge observation model", Journal of Internet Computing and Services(JICS) , Vol. 17, No. 3, pp.25-32, 2016.
<http://dx.doi.org/10.7472/jksii.2016.17.3.25>
- [4] H. Y. Choi "Computer Vision", 2013.
- [5] Nigam, Swati, Kaushik Deb, and Ashish Khare. "Moment invariants based object recognition for different pose and appearances in real scenes." Informatics, Electronics & Vision (ICIEV), 2013 International Conference on. IEEE, 2013.
<http://dx.doi.org/10.1109/ICIEV.2013.6572697>
- [6] E. Orhan, "Particle Filtering", Center for Neural Science, University of Rochester, Rochester, NY, August, 2012.
- [7] Klein, D. A., Schulz, D., Frintrop, S., & Cremers, A. B,

"Adaptive real-time video-tracking for arbitrary objects." Intelligent Robots and Systems (IROS), The 2010 IEEE/RSJ International Conference on. IEEE, pp. 772-777, Oct, 2010.

<http://dx.doi.org/10.1109/IROS.2010.5650583>

- [8] J. C. KIM, X. N. Cui, E. S. Park, H. H. Choi, H. I. Kim, "Robust PCB Image Alignment using SIFT" Journal of Institute of Control, Robotics and Systems Vol. 16, No. 7, July 2010.

<http://dx.doi.org/10.5302/J.ICROS.2010.16.7.695>

- [9] Lowe, David G. "Object recognition from local scale-invariant features." Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.

- [10] Tran, Antoine, and Antoine Manzanera. "A versatile object tracking algorithm combining Particle Filter and Generalised Hough Transform." Image Processing Theory, Tools and Applications (IPTA), November, 2015 International Conference on (pp. 105-110). IEEE. <http://dx.doi.org/10.1109/IPTA.2015.7367106>

Authors



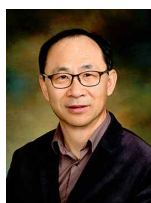
Hyoyeon Kim is a M.S. candidate student at Computer Vision Laboratory in the School of Media at Soongsil University. She received his B.S. degree in computer science from Hankyong National University, Korea, in 2015.

She is interested in computer engineering, computer vision.



Kisang Kim is a Ph.D. candidate student at Computer Vision Laboratory in the School of Media at Soongsil University. He received his B.S. (2007) and M.S. (2009) degree in computer science and the School of Media from Soongsil university.

He had work at the Daemyung Enterprise (2009~2012) and KETI (2014). Currently, he attends to several projects developing vision systems.



Prof. Choi teaches Computer Vision and Pattern. His research focuses on computer vision, pattern recognition, fuzzy inference system, and artificial intelligence. His published work deals with the smart device interface, user

interaction system, vehicle security system and helping system for blind people. Prof. Choi had served as visitor professor to the IBM Watson Research Center (1994-1995), as director to the Korea Institute of Fuzzy and Intelligent System (1996-1997) and the Korea Institute of Information Scientists and Engineers (2003~2004). As chief manager to the Culture Contents Technical Research Center in Soongsil University from 2007 to 2012, Prof. Choi sought to contribute about Automatic Signature Recognition System, Moving Object Tracking System and Adult Image Identification System. He received his B.A. (1979) in Electrical Engineering from Yonsei University and both his M.S. (1983) and Ph.D (1987) in Computer Science from the university of Michigan.