

# 피드백과 박스 보정을 이용한 Particle Filtering 객체추적 방법론

안정호\* 정회원

## Particle Filtering based Object Tracking Method using Feedback and Tracking Box Correction

Jung-Ho Ahn\* *Regular Members*

### 요 약

최근 주목을 받고 있는 Particle Filtering은 실제 객체 추적에서 발생하는 비선형, 비 가우시안 분포를 가지는 상태 벡터의 사후확률을 추정하기 위한 Monte Carlo 시뮬레이션에 기반을 둔 추적 방법론이다. 우리는 본 논문에서 Particle Filtering을 이용한 객체 추적 성능을 향상시킬 수 있는 두 가지 방법론을 제안한다. 첫 번째는 확률이 가장 낮은 샘플을 이전 프레임의 추정된 상태 벡터로 대체하는 피드백 방법론이고, 두 번째는 객체 확률 분포를 추정된 객체 후보영역에 역투영하여 신뢰구간을 구함으로써 추적 박스의 정확도를 향상시키는 방법이다. 또한, 실험을 통해 구한 추적 샘플의 진화 방정식을 제시하였다. 우리는 다양한 상황이 설정된 실험 데이터셋을 구성하여 실험을 실시하여 제안한 방법론의 우수성을 입증하였다.

**Key Words** : particle filtering, feedback, box correction, back projection, Propagation Equation, color histogram

### ABSTRACT

The object tracking method using particle filtering has been proved successful since it is based on the Monte Carlo simulation to estimate the posterior distribution of the state vector that is nonlinear and non-Gaussian in the real-world situation. In this paper, we present two novel methods that can improve the performance of the object tracking algorithm based on the particle filtering. First one is the feedback method that replace the low-weighted tracking sample by the estimated state vector in the previous frame. The second one is an tracking box correction method to find an confidence interval of back projection probability on the estimated candidate object area. An sample propagation equation is also presented, which is obtained by experiments. We designed well-organized test data set which reflects various challenging circumstances, and, by using it, experimental results proved that the proposed methods improves the traditional particle filter based object tracking method.

## I. 서론

객체 추적(Object Tracking)은 지금까지 매우 활발한 연구가 진행되고 있는 전통적인 컴퓨터 비전의 분야이다. 우주 항공, 로봇 비전, 감시 카메라 시스템 등의 분야에서 획득된 영상 시퀀스에 존재하는 객체들의 이동 위치를 자동으로 정확히 추정하는 기술은 이벤트 검출, 감시, 상호작용 등을 위해 매우 필수적이라 말할 수 있다.

영상기반 객체 추적은 3D 세계가 2D 영상으로 투영되어 발생하는 복잡한 객체의 움직임이나 부분적인 가려짐 또는

완전한 가려짐, 영상의 노이즈, 복잡한 배경, 조명 변화, 실시간 처리 등의 문제로 매우 어려운 문제로 평가받고 있다. 초기의 객체 추적 알고리즘은 객체의 움직임이 부드럽고 갑작스러운 모습의 변화가 없다는 가정 하에서 개발되었으나 최근 컴퓨터의 발달과 더불어 이러한 가정이 성립하지 않는 상황에서 좋은 성능을 보이는 알고리즘들이 많이 제시되었다 [11].

다양한 상황 변화에도 강인한 추적 알고리즘이 되기 위해 강인성, 적응성, 실시간 프로세싱 등을 고려해야해야 한다.[11] 강인성이란 복잡한 배경, 조명의 변화, 가려짐, 복잡한 객체의 움직임 등의 상황 변화에서도 관심 객체를 추적할

\*본 연구는 2011년도 강남대학교 교내연구비 지원에 의한 것임.

\*강남대학교 컴퓨터미디어정보공학부 (jungho@kangnam.ac.kr)

접수일자 : 2013년 2월 20일, 수정완료일자 : 2013년 2월 28일, 최종 게재 확정일자 : 2013년 3월 8일

수 있음을 말한다. 적응성이란 객체의 모습 변화에도 적절히 객체 모델을 업데이트 할 수 있음을 말한다. 그리고 실시간 프로세싱이란 높은 성능의 알고리즘이 빠르게 동작할 수 있도록 최적의 구현이 가능함을 말한다.

영상기반 객체 추적은 보통 불확실하고 모호한 관찰치로부터 객체의 움직임을 추론하기 위한 탐색 프로세스를 포함한다. 객체 추적에서 상태 벡터는 객체의 실제 위치와 움직임의 정보를 기술한다. 상태 벡터의 사후 분포가 가우시안 분포일 경우 Kalman 필터[9]나 확장된 Kalman 필터[10] 등의 방법론들이 최적의 솔루션을 찾기 위해 사용될 수 있다.

하지만 대부분의 실제 추적 문제는 상태 벡터의 사후 분포가 비선형적이고 비가우시안 분포를 따르기 때문에 이를 극복하기 위해 Monte Carlo 시뮬레이션에 기반을 둔 Particle Filtering 방법론[5]이 제안되었다. Particle Filtering의 주요 아이디어는 상태 벡터의 사후 분포를 가중치를 가지는 랜덤 샘플들로 표현하는 것이다.

이러한 확률적 추적 방법론은 베이저안 추정 이론(Bayesian estimation theory)에 근거하여 객체 추적의 문제를 객체 위치를 기술하는 상태 벡터의 확률밀도함수 추정의 문제로 귀결된다.[12] 이는 상태 벡터의 반복적인 측정에 근거하는데 이 추정 프로세스는 잡음(noise)에 매우 민감하게 반응한다. 이러한 이유로 확률적 추적 방법론은 보통 필터(filter)를 이용하여 상태 벡터의 실제 값을 복원하는 추정과정을 추가한다.

Particle 필터를 이용한 추적 방법론은 다중 가설(샘플)을 추적하고 상태 및 측정 프로세스의 비선형성을 효과적으로 처리할 수 있어 객체의 변화에 적응 능력이 뛰어난 객체 추적 방법론이라 할 수 있다. 이 방법론에서 다중 가설은 전과 방정식(propagation equation)에 의해 전과한다. Bimbo와 Dini[2]는 객체 추적 결과를 전과방정식의 난수 발생에 영향을 미치게 함으로써 긴 시간동안의 객체 추적을 안정화 하였다. 우리는 Particle 필터 기반 객체 추적 실험을 통해, Particle 필터 방법론이 객체의 모양이 변함에 따라 추적 박스의 모양과 크기가 적절히 변화하는 것을 확인할 수 있었다. 하지만 박스 크기의 정밀성은 높지 않음을 확인하였다.

우리는 본 논문에서 다중 가설의 전과 방정식과 추적 박스 보정 방법을 제안하였고, 추정 결과를 피드백하여 샘플을 업데이트하는 방식을 소개하고, 실험을 통해 추적 성능을 향상을 확인할 수 있었다. 본 논문의 우리의 이전 결과[1]를 확장한 것이다.

우리는 2절에서 컬러특징과 Particle Filtering 방법론을 이용한 객체 추적 방식과 실험을 통해 구한 전과방정식을 소개한다.

## II. Particle Filtering을 이용한 객체 추적

우리는 본 절에서 Particle Filtering을 이용한 객체 추적

방법을 소개한다. 제안하는 방법론은 Nummiaro et. al[7]이 제안한 컬러 기반 Particle Filtering 방법론에 기반을 두고 있으며, 전과 방정식과 상태 벡터 피드백을 이용하는 방법론을 새롭게 제안한다.

### 1. 컬러 모델

초기에 주어진 객체 영역과 샘플 영역을 RGB 컬러 히스토그램으로 모델링한다. RGB 각 채널은 8개의 Bin(Bin)으로 양자화 하였으며, 추적 객체의 외곽 박스(Bounding Box)의 경계부분은 배경이 포함될 확률이 높으므로 다음과 같은 커널(kernel)을 이용하여 각 픽셀에 가중치를 부여하여 히스토그램의 빈값에 누적하였다[7].

$$k(r) = \begin{cases} 1 - r^2 & (r < 1) \\ 0 & (r \geq 1) \end{cases} \quad (1)$$

이 커널을 사용하여 중심위치가  $C$  인 박스의 히스토그램  $p = \{p_k\}_{k=1}^m$  는 다음과 같이 구할 수 있다.

$$p_k = K \sum_{i=1}^I k \left( \frac{\|C - x_i\|}{\sqrt{H_x + H_y}} \right) \delta(h(x_i) - k) \quad (2)$$

여기서  $I$ 는 박스내의 픽셀수,  $K$ 는 정규화 상수,  $\|\cdot\|$ 는  $L^2$ 노름,  $H_x, H_y$ 는 박스의 가로, 세로 길이의 절반이며,  $\delta(\cdot)$ 는 0 또는 1값을 가지는 Kronecker 델타함수,  $h(x_i)$ 는  $i$  번째 픽셀  $x_i$ 의 RGB 컬러값의 해당 1차원 인덱스로 0과  $8^3 - 1 (= 511)$  사이의 값이다. 정규화 상수  $K$ 는  $\sum_{k=0}^{511} p_k = 1$  을 만족하기 위해 다음과 같이 주어진다.

$$K = \frac{1}{\sum_{i=1}^I k \left( \frac{\|C - x_i\|}{\sqrt{H_x + H_y}} \right)} \quad (3)$$

추적 샘플들의 가중치를 객체 컬러 히스토그램과 샘플 영역의 컬러 히스토그램간의 유사도로 정의하기 위해 주어진 두 히스토그램  $p = \{p_k\}_{k=1}^m$  와  $q = \{q_k\}_{k=1}^m$  사이의 유사도를 다음과 같이 Bhattacharyya 계수로 정의한다[6].

$$\rho[p, q] = \sum_{k=1}^m \sqrt{p_k q_k} \quad (4)$$

$\rho$  값이 높을수록 두 히스토그램은 유사함을 알 수 있다. 따라서 우리는 두 히스토그램 분포간의 거리로 다음과 같은 Bhattacharyya 거리를 사용한다.

$$d(p, q) = \sqrt{1 - \rho[p, q]} \quad (5)$$

## 2. 객체 추적을 위한 Particle Filtering 방법론

우리는 이 절에서 Particle 필터를 이용한 추적 방법론을 재 기술하고 실험을 통해 구한 전과 방정식을 소개한다.

벡터  $X_t$  를 추적 객체의 상태(state)를 기술하기 위한 상태 벡터이고, 벡터  $Z_t$  를 시간  $t$  까지의 관찰치들이라 할 때, Particle 필터는 사후 확률  $p(X_t | Z_t)$  를 가중치를 가진 샘플(가설) 집합  $S$  에 의해 근사시키는 것이다.

$$S = \{(s^{(n)}, \pi^{(n)}) | n = 1, \dots, N\} \quad (6)$$

여기서  $n$  번째 샘플  $s^{(n)}$  는 추적 객체에 대한 하나의 가설(hypotheses)로 가중치  $\pi$  를 가진다. 우리는 샘플  $s$  를 다음과 같이 모델링하였다.

$$s = (x, y, \dot{x}, \dot{y}, H_x, H_y, \dot{h}) \quad (7)$$

여기서  $x, y$  는 박스의 중심위치이고,  $H_x, H_y$  는 박스의 가로, 세로 길이의 절반이며,  $\dot{x}, \dot{y}$  는 박스의  $x, y$  방향 변화율,  $\dot{h}$  는 박스 크기 변화율을 의미한다.

매 프레임마다 복원 추출로  $N$  개의 샘플을 추출하고, 추출된 각 샘플은 일정한 전과 방정식(propagation equation)에 의해 진화한다.

진화한 각 샘플에 대해 히스토그램 유사도를 이용하여 관찰 확률  $\pi^{(n)} = p(z_t | X_t = s_t^{(n)})$  을 다음과 같이 업데이트한다.

$$\pi_t^{(n)} = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(1-d(p_{s^{(n)}}), q))}{2\sigma^2}} \quad (8)$$

여기서 지수에 있는  $d(p_{s^{(n)}}, q)$  는 샘플 영역의 컬러 히스토그램  $p_{s^{(n)}}$  과 객체의 컬러 히스토그램  $q$  의 Bhattacharyya 거리이다. 이 업데이트한 가중치를 이용하여 추적 객체에 대한 평균 상태 벡터를 다음과 같이 추정한다.

$$E(S) = \sum_{n=1}^N \pi^{(n)} s^{(n)} \quad (9)$$

이 평균 상태 벡터는 현 프레임에서 추정된 객체의 위치를 의미한다. Particle 필터 방법론은 매 프레임 다중 가설을 동시에 고려함으로써 불확실성을 모델링하는 강인한 추적 방식이라고 할 수 있다. 우리는 각 샘플들을 실험을 통해 구한 다음 전과 방정식을 이용하여 전파시켰다.

$$\dot{x}_t = x_{t-1} + 0.05 \dot{x} + \epsilon H_x + M \quad (10)$$

$$\dot{y}_t = y_{t-1} + 0.05 \dot{y} + \epsilon H_y + M \quad (11)$$

$$\dot{x}_t = x_{t-1} + \epsilon \quad (12)$$

$$\dot{y}_t = y_{t-1} + \epsilon \quad (13)$$

$$(H_x)_t = (H_x)_{t-1} + \dot{h}(H_x)_{t-1} + 10\epsilon + M \quad (14)$$

$$(H_y)_t = (H_y)_{t-1} + \dot{h}(H_y)_{t-1} + 10\epsilon + M \quad (15)$$

$$\dot{h}_t = h_{t-1} + 0.001\epsilon \quad (16)$$

여기서  $\epsilon$  은 평균이 0이고, 분산이 0.5 인 정규분포에서 생성된 확률변수이며 마진  $M$  은 실험에서 0.5로 설정하였다.

## 3. 피드백을 이용한 샘플 업데이트

Particle Filtering은 상태 벡터  $X_t$  의 사후 확률을 샘플들로 묘사하는 방법론이다. 이 샘플들을 통해 비 가우시안 분포를 가지는 상태벡터의 분포를 근사할 수 있는 것이다.

낮은 가중치를 가지는 샘플들은 객체의 급작스러운 변화를 감지할 수 있기에 의미가 있다. 하지만 Particle Filtering 방법론은 보통 많은 샘플을 사용하기 때문에 가장 낮은 가중치를 가지는 샘플은 객체의 위치를 정확히 추적하는 데 큰 도움이 되지 못 할 뿐 아니라, 오히려 식(9)의 가중 평균으로 객체의 최종 위치를 추정하는 과정에서 정확한 상태 벡터 추정에 안 좋은 영향을 끼칠 가능성이 높다.

반면, 추정된 상태 벡터는 객체의 움직임이 크지 않을 때 다음 프레임의 객체 위치와 매우 유사할 가능성이 높다. 이러한 이유로 우리는 가중치가 가장 낮은 샘플을 추정된 상태 벡터로 대체하는 피드백 방식을 고안하였다. 피드백을 이용한 Particle Filtering 추적 방법은 다음과 같다.

1. 현재 프레임에서 식 (9)를 이용하여 상태 벡터를 추정한다.
2. 가장 낮은 가중치를 가지는 샘플  $s$  를 찾아 샘플 집합에서 제외시킨다.
3. 식(8)을 이용하여 상태 벡터의 가중치를 구한다.
4. 위에서 구한 가중치와 추정된 상태 벡터를 샘플 집합에 추가한다.

## III. 추적 박스 보정

Particle 필터를 이용한 객체의 추적은 다중 가설(샘플)을 사용하기 때문에 열악한 추적 상황에서도 다른 추적 방법론보다 강인할 수 있다는 장점이 있다. 샘플의 가중치 평균을 이용하여 객체의 박스를 결정하는 과정에서 박스의 크기와 모양이 적응적으로 변하지만, 추적 박스의 추정의 정확성이

다소 떨어진다는 단점이 있다. 이러한 이유로 우리는 박스 모양과 크기의 정확성을 높일 수 있는 후처리 방법론을 개발하였다.

Particle 필터에 의해 구해진 박스의 중심 위치가  $(x_c, y_c)$  이고  $H_x$  와  $H_y$  를 각각 가로 길이와 세로 길이의 반이라 할 때, 후처리를 위한 후보 박스  $B$  를 다음과 같이 설정한다.

$$[x_c - kH_x, x_c + kH_x] \times [y_c - kH_y, y_c + kH_y] \quad (17)$$

여기서  $k$  는 후보영역의 크기를 결정하는 상수로 실험에서는 2 를 사용하였다. 이 사각형 영역의 각 픽셀에 대해 객체 컬러 히스토그램의 값을 투영하여 행과 열의 히스토그램의 분포를 다음과 같이 조사한다. 즉,  $W \times H$  해상도의 영상에 대하여 열 히스토그램을  $q^c = \{q_i^c\}_{i=1}^W$  은 다음과 같이 주어진다.

$$q_j^c = \sum_{i=1}^H p(x_{i,j}) \delta((i,j) \in B) \quad (18)$$

여기서  $p(\cdot)$  는 객체의 히스토그램이고  $x_{i,j}$  는  $i$  행  $j$  열 픽셀의 컬러 인덱스 값,  $\delta(\cdot)$  는 Kronecker 델타함수이다. 열 히스토그램의  $100(1 - \alpha)\%$ 에 해당하는 영역을 취하기 위하여 우리는 다음과 같이 최종 추적 박스의 가로 방향 폭을 결정한다.

$$y_{\min} = \min \left\{ 1 \leq j \leq W \mid \sum_{i=1}^j q_i^c / \sum_{i=1}^W q_i^c > \alpha \right\} \quad (19)$$

$$y_{\max} = \max \left\{ 1 \leq j \leq W \mid \sum_{i=1}^j q_i^c / \sum_{i=1}^W q_i^c < 1 - \alpha \right\} \quad (20)$$

마찬가지로, 행 히스토그램  $q^r = \{q_i^r\}_{i=1}^H$  에 대하여 다음과 같이 추적박스의 세로 방향 폭을 결정한다.

$$x_{\min} = \min \left\{ 1 \leq i \leq H \mid \sum_{j=1}^i q_j^r / \sum_{j=1}^H q_j^r > \alpha \right\} \quad (21)$$

$$x_{\max} = \max \left\{ 1 \leq i \leq H \mid \sum_{j=1}^i q_j^r / \sum_{j=1}^H q_j^r < 1 - \alpha \right\} \quad (22)$$

여기서  $\alpha$  는 신뢰도를 의미하는 상수로 실험에서 0.05 를 사용하였다. 이렇게 구해진  $x, y$  축 경계를 이용하여 추적 박스를 다음과 같이 보정한다.

$$[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \quad (23)$$

다음 [그림 2]에서 왼쪽 그림의 노란색 박스는 Particle

Filtering 방법론에 의해 구해진 박스이고 하얀색 박스는 보정된 추적 박스를 보여준다. 오른쪽 그림에서 경계 부분의 하얀색 영역은 열 및 행 히스토그램을 의미하며 회색 영역은 히스토그램 역투영 영역을 보여준다. 실선은 박스 보정을 위한 90% 신뢰영역을 보여 준다.



그림 6. 역투영에 의한 박스 보정 방법

### III. 실험 및 분석

우리는 제안한 방법론의 정확도를 평가하기 위해서 대표적인 객체 추적 방법론과 비교 실험을 수행하였을 뿐 아니라 제안한 추가 방법론의 효율성을 보이기 위한 실험도 수행하였다. 실험을 위해 테스트 데이터는 3가지 종류의 객체에 대해 각각 7개의 테스트 비디오를 촬영하였다. 객체의 종류는 단색 객체(S), 다색 객체(M), 얼굴(F) 등 세 가지이고, 각 객체에 대하여 7가지 특성을 가지는 테스트 비디오를 각각 촬영하였다. 각 비디오의 특성은 다음 <표 1>과 같다.

표 1. 테스트 데이터의 특성

비디오 No.	특성
1	객체 크기 변화
2	빠른 움직임
3	2D 회전
4	3D 회전
5	많이 가려짐
6	조금 가려짐
7	일시적인 사라짐

비디오의 첫 프레임에서 객체 영역을 마우스로 사각형 모양으로 지정한 후 다음 프레임부터 수동으로 추출한 실 객체 영역(Ground Truth)와 비교하여 정확도를 측정하였다.  $t$  번째 프레임의 정확도는 추적기(tracker)에 의해 추정된 박스의 중심이 실 객체영역에 속하면 성공, 그렇지 않으면 실패로 규정하여 점 추정 정확도 P를 구하였으며, 박스 크기의 정확도를 측정하기 위해 Phillips and Chhabra[8]가 제안한 다음 식에 의해 박스의 정확도(Q)를 추정하였다.

$$Q_t(B_t^G, B_t^E) = \frac{|B_t^G \cap B_t^E|}{|B_t^G \cup B_t^E|} \quad (24)$$

여기서  $Q_t$ 는  $t$  번째 프레임의 추적성능으로 실 객체의 박

표 2 박스보정 알고리즘의 비교 실험 결과

	S		M		F		Total	
	P	Q	P	Q	P	Q	P	Q
PF(F,B)	0.88(0.16)	0.36(0.10)	0.71(0.30)	0.39(0.17)	0.83(0.14)	0.41(0.10)	0.81(0.20)	0.39(0.12)
PF(B)	0.86(0.18)	0.34(0.11)	0.71(0.30)	0.38(0.18)	0.82(0.14)	0.39(0.10)	0.80(0.22)	0.37(0.13)
PF	0.84(0.19)	0.31(0.13)	0.71(0.31)	0.37(0.15)	0.78(0.16)	0.34(0.11)	0.77(0.24)	0.34(0.13)
MS	0.68(0.19)	0.34(0.10)	0.13(0.14)	0.11(0.06)	0.75(0.21)	0.36(0.09)	0.52(0.33)	0.27(0.14)
CS	0.75(0.30)	0.39(0.18)	0.05(0.07)	0.04(0.03)	0.66(0.33)	0.32(0.17)	0.49(0.40)	0.25(0.21)

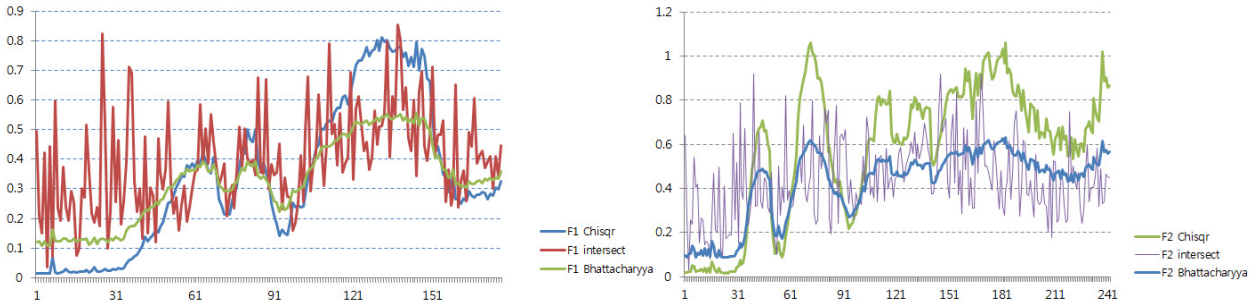


그림 7. 히스토그램 거리 측정방법의 정확성 비교

스영역  $B_t^G$ 와 추정된 박스영역  $B_t^E$ 의 겹침의 정도를 측정하는 것으로 일반적으로 작은 값을 갖는 특징이 있다.

우리는 객체 추적을 위한 히스토그램의 거리 함수의 적합성을 측정하기 위해 카이제곱( $\chi^2$ ) 거리과 교차(Intersection) 거리, Bhattacharyya 거리를 비교하였다. 비교 실험을 위해 매 프레임 실 객체 영역에 대해 컬러 히스토그램을 구한 후 첫 프레임에서 구한 객체 컬러 히스토그램과의 거리를 비교하였다. 다음 그림 3의 그래프는 두 개의 테스트 비디오 셋에 대한 실험 결과를 보여 준다. x 축은 프레임을 의미하며 y 축은 실 객체 영역의 거리(dissimilarity) 값이다. 그래프에서 볼 수 있듯이 교차 거리는 매우 불안정하며, 평균적으로 보았을 때 우리가 실험에서 사용한 Bhattacharyya 거리가 카이제곱 거리 보다 우수한 성능을 보임을 확인할 수 있다.

우리는 제안한 방법론들의 성능을 비교 평가하기 위하여 컬러특징과 Particle 필터를 이용한 방법 PF[7]와 Mean Shift[4], CAM-Shift[3] 등 3가지 방법론과 PF에 박스 보정을 추가한 알고리즘 PF(B), FP(B)에 피드백 알고리즘을 추가한 FP(F, B)을 준비된 21(= 3 객체 × 7 개의 비디오)개의 테스트 데이터 셋에 대해 실험하였다. <표 2>는 3개의 객체 별 실험 결과를 보여 준다. 박스 정확도 Q는 식 (17)의  $Q_i$ 의 평균값이며 괄호 안의 숫자는 표준편차를 의미한다.

이 다섯 가지 방법론은 첫 프레임에서 지정된 객체영역의 정확도에 의해 성능이 좌우될 수 있고, Particle 필터는 확률과정(stochastic process)에 의존하고 있기 때문에 우리는 각 테스트 데이터에 대하여 5번의 반복 실험을 수행하여 정확도를 측정하였다.

표 2는 실험에서 나타난 정확도 결과를 보여 준다. Mean

Shift(MS) 방법론과 CAM-Shift(CS) 방법론의 결과는 데이터 셋의 난이도를 알려주는 참고 지표로 볼 수 있다. 제안한 박스 보정 방법론 PF(B)는 기존의 Particle 필터링 방법론 PF보다 박스의 정확도가 향상되었음을 확인할 수 있고, PF(F, B)는 모든 데이터 셋에 대해 중점 정확도 P와 박스 정확도 Q에서 모두 가장 우수한 성능을 보였다.

표 3. 알고리즘 별 속도 측정 결과(단위: fps)

	F	S	M	Total
PF(F, B)	53.07	62.13	20.72	36.06
PF(B)	48.10	53.59	21.11	34.55
PF	76.05	80.52	25.56	46.37
MS	55.31	53.85	53.71	54.28
CS	54.17	54.00	52.60	53.58

표 3은 위 5개의 알고리즘의 속도 측정 결과를 보여준다. 알고리즘의 속도는 객체 크기에 비례하여 감소하는 경향이 있으므로 테스트 셋에 대하여 다르게 나타남을 볼 수 있다. 제안한 박스 보정 방법론 PF(B)는 기존 알고리즘 PF에 후처리 알고리즘을 추가하였기 때문에 기존 알고리즘 보다 느리지만 무난히 실시간 프로세싱이 가능함을 확인할 수 있다. 특히하게도 더 많은 프로세싱을 하도록 디자인된 PF(F, B) 방법론이 PF(B) 방법론보다 더 빠른 처리 속도를 보였다. 이는 가중치가 낮은 샘플은 보통 객체 크기와 무관하게 넓은 영역을 차지할 가능성이 있기 때문인 것으로 해석된다.

그림 4는 왼쪽부터 차례로 CS, MS, PF, PF(F, B)의 객체 M에 대한 실험 결과를 보여 준다. 제안한 방법론의 박스가 추적 객체인 책과 매우 유사함을 볼 수 있다.

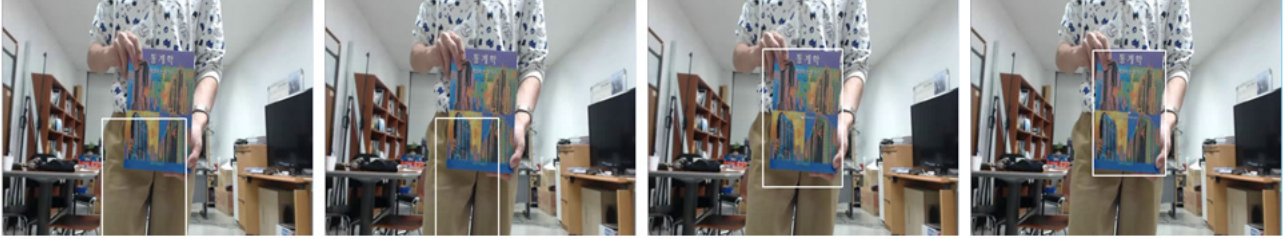


그림 8. 객체 추적 비교 실험 결과

#### IV. 토의 및 결론

우리는 본 논문에서 전통적인 Particle Filtering을 이용한 객체 추적 방법론을 향상시킬 수 있는 두 가지 방법론을 제시하였다. 제안된 박스 보정 방법은 추정된 추적 박스 근방에서 객체 확률 분포를 영상에 역투영하여 확률의  $100(1-\alpha)\%$  신뢰구간을 구하는 방식으로 구성하였으며, 각 프레임에서 확률(가중치)이 가장 낮은 샘플을 객체 상태 추정 결과로 대체함으로써 성능과 속도 면에서 향상을 가져올 수 있었다.

제안한 방법론은 실시간 데모에서 추적 객체의 크기 변화를 비교적 정확하게 추정하였을 뿐 아니라, 추적 객체가 일시적으로 사라지고 다시 출현하는 상황에서도 객체를 놓치지 않고 추적할 만큼 안정적인 성능을 보였다. 진화하는 랜덤 샘플들을 이용한 Particle Filtering 추적 방법론은 일정한 윈도우 영역을 모두 탐색하는 추적 방법론에 비해 효율이 높고, 객체의 갑작스러운 움직임에도 강인하게 추적함을 많은 실험을 통해 확인할 수 있었다.

객체 추적은 보통 추적 객체의 움직임 분석을 통해 영상에서 일어나는 현상을 감지하기 위해 수행된다. 이러한 경우 객체 추적 알고리즘은 매우 빠르게 동작할 수 있어야 한다. 우리는 앞으로 속도 향상을 위한 연구를 수행할 계획이다.

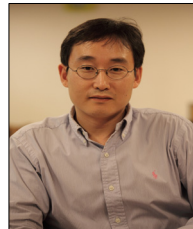
#### 참 고 문 헌

[1] 안정호, 강봉, 황인욱, “추적박스 보정을 이용한 향상된 Particle Filter 객체 추적 방법론”, 멀티미디어학회 추계학술발표대회 논문집, 15권 2호, pp. 355-358, 2012.  
 [2] A. D. Bimbo and F. Dini, “Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation”, Computer Vision and Image Understanding, vol. 115, no. 6, pp. 771-786, 2011.  
 [3] G.R. Bradski. “Computer vision face tracking as a component of a perceptual user interface”, In Workshop on Applications of Computer Vision, pp. 214-219, 1998.  
 [4] D. Comaniciu, V. Ramesh and P. Meer, “Real-Time Tracking of Non Rigid Objects using Mean Shift”, International Conference on Computer Vision and Pattern Recognition, pp. 70-73, 2000.

[5] M. Isard and A. Blake, “CONDENSATION - Conditional Density Propagation for Visual Tracking”, International Journal on Computer Vision, Vol. 1, No. 29, pp. 5-28, 1998.  
 [6] T. Kailath, “The Divergence and Bhattacharyya Distance Measures in Signal Selection”, IEEE Transactions on Communication Technology COM, vol. 15, no.1, pp.52-60, 1967.  
 [7] K. Nummiaro, E. Koller-Meier and L. V. Gool, “An Adaptive Color-Based Particle Filter” Image and Vision Computing, Vol. 21, pp. 99-110, 2002.  
 [8] I. T. Phillips and A. K. Chhabra, “Empirical performance evaluation of graphics recognition systems”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 9, pp. 849-870, 1999.  
 [9] Y. Bar-Shalom and T. Fortmann, Tracking and data association, Academic Press, 1988.  
 [10] G. Welch and G. Bishop, “An Introduction to Kalman filter”, Technical Report (TR 95-041), University of North Carolina at Chapel Hill, 2004.  
 [11] H. Yang, L. Shao, F. Zheng, L. Wang and Z. Song, “Recent advances and trends in visual tracking: A review”, Neurocomputing, vol.74, pp. 3823-3831, 2011.  
 [12] A. Yilmaz, O. Javed and M. Shah, “Object tracking: A survey”, ACM Journal of Computing Surveys, vol. 38, no. 4, 2006.

#### 저자

안정호(Jung Ho Ahn)



· 2007년 3월~현재 : 강남대학교 컴퓨터미디어정보공학부 교수  
 <관심분야> : 컴퓨터비전, 패턴인식, 제스처인식, 얼굴인식

정희원

- 1996년 2월 : 연세대학교 수학과 학사 졸업
- 1998년 2월 : 연세대학교 수학과 석사 졸업
- 2006년 8월 : 연세대학교 컴퓨터과학과 박사졸업