

실시간 객체 추적을 위한 Condensation 알고리즘과 Mean-shift 알고리즘의 결합

조 상 현^{*} · 강 행 봉^{**}

요 약

실시간 객체 추적(Real-time object tracking)은 비디오 감시 시스템, 비전 기반 네비게이터와 같은 비전 응용 산업이 발달하면서 그 중요성이 더해지고 있는 분야이다. 객체 추적을 위해 많이 이용되고 있는 알고리즘으로 Mean-shift와 Condensation 알고리즘이 있다. Mean-shift 알고리즘을 기반으로 한 객체 추적 알고리즘은 구현이 간단하고, 적은 계산 복잡도를 갖는 장점이 있다. 따라서 실시간 객체 추적 시스템에 적합하다고 할 수 있지만, 지역 모드(Local mode)로 수렴하는 특성으로 인해 복잡한 환경(Cluttered environment)에서는 좋은 성능을 나타내지 못하는 단점을 가지고 있다. 반면, 여러 개의 후보들을 이용해 객체의 위치를 추정하는 Condensation 추적 알고리즘은 복잡한 환경에서 특정 객체를 추적하는데 많이 사용된다. 하지만 Condensation 알고리즘을 기반으로 한 추적 알고리즘은 정확한 추적을 하기 위해서 복잡도가 높은 객체 모델과 많은 수의 후보가 요구된다. 따라서 높은 복잡도를 갖게 되고, 이것으로 인해 복잡한 환경에서는 실시간 구현이 어렵다는 단점을 갖게 된다.

본 논문에서는, 복잡한 환경에서 실시간 객체 추적에 적합하도록 Condensation 알고리즘과 Mean-shift 알고리즘을 결합해서, 적은 수의 후보들을 이용하는 모델을 제안한다. 적은 수의 후보들을 이용하더라도, Mean-shift 알고리즘을 이용해 보다 높은 유사도를 가지는 후보들만을 이용함으로써, Condensation 알고리즘이나 Mean-shift 알고리즘만을 이용할 때보다 더 나은 성능을 얻을 수 있었다.

키워드 : Mean-shift, Condensation, Particle Filter, 객체 추적

Integration of Condensation and Mean-shift algorithms for real-time object tracking

Sang-Hyun Cho^{*} · Hang-Bong Kang^{**}

ABSTRACT

Real-time Object tracking is an important field in developing vision applications such as surveillance systems and vision based navigation. mean-shift algorithm and Condensation algorithm are widely used in robust object tracking systems. Since the mean-shift algorithm is easy to implement and is effective in object tracking computation, it is widely used, especially in real-time tracking systems. One of the drawbacks is that it always converges to a local maximum which may not be a global maximum. Therefore, in a cluttered environment, the Mean-shift algorithm does not perform well. On the other hand, since it uses multiple hypotheses, the Condensation algorithm is useful in tracking in a cluttered background. Since it requires a complex object model and many hypotheses, it contains a high computational complexity. Therefore, it is not easy to apply a Condensation algorithm in real-time systems.

In this paper, by combining the merits of the Condensation algorithm and the mean-shift algorithm, we propose a new model which is suitable for real-time tracking. Although it uses only a few hypotheses, the proposed method use a high-likelihood hypotheses using mean-shift algorithm. As a result, we can obtain a better result than either the result produced by the Condensation algorithm or the result produced by the mean-shift algorithm.

Key Words : Mean-shift, Condensation, Particle Filter, Real-time Tracking

1. 서 론

객체 추적(Object tracking)은 비디오 감시 시스템 뿐만

아니라, 화상 회의 시스템, 운전자 보조 시스템과 같은 여러 다양한 비전 응용분야에서 그 중요성이 매우 높다. 일반적으로 객체 추적 방법은 크게 Bottom-up 방식과 Top-down 방식 2가지가 있다. Bottom-up 방식은 이미지로부터 객체를 분리하는 과정을 통해 추적을 하는 방법이다. 객체와 배경을 분리하는 방법이 간단하기 때문에 많이 사용되지만 이

^{*} 준 회 원 : 가톨릭대학교 컴퓨터정보공학과 박사과정

^{**} 정 회 원 : 가톨릭대학교 컴퓨터정보공학과 교수

논문접수 : 2004년 7월 20일, 심사완료 : 2005년 3월 30일

방법은 배경 모델을 가지고 있어야 하고, 조명의 변화나 겹침(Occlusion)이 발생하는 복잡한 환경에 대해서는 적용하기가 어렵다는 단점이 있다. 한편, Top-down 방식은 여러 객체 후보들을 발생시킨 뒤에 발생된 후보 객체들을 확인하는 과정을 통해서 객체를 추적하는 방법이다. 이러한 Top-down 방식 중에 많이 이용되는 방법이 Condensation 알고리즘[1-2]과 Mean-shift 알고리즘[3-5]이다.

Condensation 알고리즘[1-2]을 이용한 기법은 Factored sampling 기법에 기반한 확률적인 객체 추적 기법으로 복잡한 상황에서 특정 객체를 추적하는데 많이 사용된다. 하지만 Condensation 알고리즘을 이용한 기법은 정확한 추적을 하기 위해서 복잡도가 높은 객체 모델과 많은 수의 후보가 필요하게 된다. 따라서 실시간 추적에 적용할 경우, 높은 복잡도를 갖게 되어, 실시간 구현이 어렵다는 단점을 갖게 된다.

한편, Mean-shift 알고리즘을 이용한 추적 기법은 구현이 간단하고, 단일 후보를 이용하기 때문에 Condensation 알고리즘에 비해 상대적으로 적은 계산 복잡도를 가지고 있다는 장점이 있다. 하지만 지역 모드로 수렴하는 특성으로 인해 객체 사이의 겹침과 같은 상황에서는 추적이 실패하게 된다.

본 논문에서는 복잡한 환경에서 실시간 객체 추적에 적합하도록 Condensation 알고리즘과 Mean-shift 알고리즘을 결합해서, 적은 수의 후보들을 이용하는 모델을 제안한다. 제안한 모델은 Condensation 알고리즘에 비해 상당히 적은 후보를 이용하게 되는데, 사용되는 후보들의 유사도가 Mean-shift 알고리즘에 의해 비교적 높아지기 때문에, Condensation 알고리즘에 비해 비효율적인 계산을 상당히 줄일 수 있다. 뿐만 아니라, 복수개의 후보를 이용함으로써, Mean-shift가 가지는 단점도 보완할 수 있었다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해서 서술하고, 3장에서는 본 논문에서 사용하는 Condensation 알고리즘과 Mean-shift 알고리즘에 대해 서술하고, 4장에서는 제시하는 추적모델에 대해서 설명한다. 5장에서는 제안된 시스템의 실험결과에 대해 살펴보고, 마지막으로 6장에서는 결론을 기술한다.

2. 관련 연구

객체 추적(Object Tracking)에 관해서 가장 널리 사용되고 있는 것은 칼만 필터(Kalman filter), Mean-shift, Condensation 알고리즘이다. 칼만 필터란 잡음(noise)에 의해 간섭받는 선형 동적 시스템(Linear dynamic system)에서 상태 x_t 의 추정치 \hat{x}_t 를 구하기 위한 재귀적인(Recursive) 알고리즘이다. 타겟의 상태는 동적 방정식(Dynamic Equation) $x_k = f_k(x_{k-1}, v_k)$ 에 의해서 정의되며, 상태에 대응되는 측정치(Measurement)는 측정 방정식(Measurement Equation) $z_k = h_k(x_{k-1}, n_k)$ 에 의해서 정의된다. 여기서 v_k 와 n_k 는 잡음으로서 모두 평균이 0인 가우시안(Gaussian) 잡음이다.

칼만 필터는 크게 3가지 단계로 이루어져 있다. 처음에는 예측(Prediction)의 단계로서 다음 상태와 상태 추정 오차의 공분산(Covariance) 행렬을 예측하는 단계이다. 두번째 단계는 측정(Measurement) 단계로서 측정행렬(Measurement Matrix)과 예측단계에서 예측된 다음 상태값, 측정 오차값을 이용해서 측정값을 구하는 단계이다. 마지막은 갱신(Update) 단계로서 측정 단계에서 구해진 측정값을 이용해서 상태와 상태 추정 오차를 갱신하는 단계이다. 여기서 만약 f_k 와 h_k 가 비선형이라면, 확장된 칼만 필터(Extended Kalman Filter)가 사용된다.

Boykov와 Huttenlocher[7]는 칼만 필터를 이용해서 차량 추적(Vehicles tracking)에 관한 연구를 했으며, Rosales와 Sclaroff[8]는 2차원 이미지 모션으로부터 3차원 객체 궤적(trajjectory)를 추정하기 위해 EKF(Extended Kalman Filter)를 이용했다.

Comaniciu, Meer, Ramesh [3-5]는 Fukunaga, Hostetler [9]에 의해 제안된 Mean-shift를 이용해서 칼라 분포(Color distribution)로 표현되어지는 다중 후보들의 칼라분포(Color distribution)의 유사도(similarity)를 계산해서 객체 추적을 하는 Mean-shift 추적기를 제안했다. 단일 후보를 이용하기 때문에 비교적 적은 복잡도를 가지는 장점이 있지만 지역 모드(Local Mode)로 수렴하는 특성으로 인해 겹침과 같은 상황에서는 추적이 실패하는 단점이 있다. Collins[10]은 이러한 Mean-shift를 이용해 추적기의 커널(Kernel)의 크기를 적절하게 변화시키는 추적방법을 제안했다.

Isard[1-2] 등은 GMM(Gaussian Mixture Model)과 Condensation 알고리즘을 이용한 추적기 모델을 제시했고, Perez [11] 등은 다중 객체 추적(Multiple object tracking)을 위해 칼라 히스토그램(Color histogram)과 Condensation 알고리즘을 이용했다. Nummiaro[11-12]등은 칼라분포(Color distribution)와 Condensation 알고리즘을 이용한 칼라 변화에 적응적인 추적기 모델을 제시했다. 두 가지 모델의 경우 모두 복잡한 환경에 대해서 강인한 추적 성능을 보여주었지만, 많은 후보를 사용해야 하기 때문에 높은 계산 복잡도가 요구되는 단점을 가지고 있다.

3. Condensation 알고리즘과 Mean-shift를 이용한 객체 추적 방법

3.1 객체의 표현

본 논문에서 객체는 타원으로 정의한다. 따라서, 객체의 상태를 나타내는 벡터 x 를 다음과 같이 정의한다.

$$\mathbf{x} = \{x, y, \dot{x}, \dot{y}, H_x, H_y, \dot{a}\} \quad (1)$$

여기서 x, y 는 타원의 중심, \dot{x}, \dot{y} 는 모션(motion), H_x, H_y 는 각각 타원의 장축과 단축의 길이의 절반, a 는 크기의 변

화도를 나타낸다.

칼라 분포(Color distribution)는 객체의 non-rigidity, 회전(rotation) 등에 강인하기 때문에 타겟의 모델로 사용된다. 경계(Boundary) 픽셀이 배경에 속하거나 겹쳤을 때, 칼라 분포의 신뢰도를 높이기 위해 타원의 중심으로 멀어질수록 적은 가중치(weight)를 갖도록 다음과 같은 가중치 함수를 도입한다.

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

여기서, r 은 타원의 중심으로부터의 거리이다.

이 가중치 함수를 이용해서, 중심이 y 이고, 가로축 H_x 와 세로축 H_y 를 갖는 타원의 칼라분포는 다음과 같이 계산된다.

$$p_y = f \sum_{i=1}^l k\left(\frac{\|y - x_i\|}{a}\right) \delta[h(x_i) - u] \quad (3)$$

여기서, k 는 커널 프로파일 함수(kernel profile function), l 는 타원 안의 pixel의 수, δ 는 Kronecker delta 함수, a 는 $\sqrt{H_x^2 + H_y^2}$, f 는 normalization factor이다.

3.2 Condensation 알고리즘

Condensation(Bayesian Filter, Monte Carlo Method, Particle filter) 알고리즘은 Factored Sampling에 기반한 확률적인 객체 추적 알고리즘이다[1-2]. 시간 t 에서의 객체의 상태를 벡터 x_t , 그 때의 관측치(Observation)을 z_t 라고 하자.

Condensation 알고리즘은 다음과 같이 요약할 수 있다.

① 샘플링(Sampling)

이전 단계의 Conditional Observation 확률 $p(x_{t-1} | z_{t-1})$ 을 이용해서 후보 집합 $\{s_{t-1}, \pi_{t-1}\}$ 로부터 샘플링해서 새로운 후보 집합 S'_{t-1} 를 얻는다.

② 예측(Prediction)

확률적 동적 모델 $p(x_t = s_t | x_{t-1} = s_{t-1})$ 를 이용해 새로운 후보 s_t 를 예측한다.

③ 측정(Measurement)

모든 후보에 대해서 다음을 계산한다.

$$\pi_t^i = p(z_t | x_t = s_t^i)$$

다음 과정으로 프레임의 샘플링 단계로 다시 넘어간다.

여기서 π_t^i 는 시간 t 의 i 번째 후보의 가중치를 나타낸다.

이 가중치가 높을수록 이 후보가 다음 샘플링 단계에서

선택될 확률이 높아진다.

Condensation 알고리즘은 복수개의 후보를 이용하기 때문에 복잡한 배경에서 추적이 용이하고, 또한 일시적인 오류가 발생하더라도 복구가 가능한 장점이 있는 반면, 계산 복잡도가 높아지는 단점이 있다.

3.3 Mean-shift 알고리즘

Mean-shift 알고리즘을 이용한 추적 알고리즘은 현재의 위치벡터 x 로부터 식(4)에 의해 계산되는 Mean-shift 벡터 방향의 새로운 위치 벡터 $x' = x + \Delta x$ 을 계산해내는 방법이다[12].

$$\Delta x = \frac{\sum_a K(a-x)w(a)a}{\sum_a K(a-x)w(a)} \quad (4)$$

여기서, K 는 커널(kernel)함수이고, x 는 현재 위치, a 는 객체 영역에 속하는 픽셀의 개수, w 는 가중치(weight)함수이다.

Mean-shift을 이용한 칼라 기반 추적 알고리즘은 다음과 같이 요약할 수 있다[3-5].

앞 프레임에서의 타겟모델 $\{\hat{q}_u\}_{u=1, \dots, m}$ 과 그것의 위치 \hat{y}_0 는 정해져 있다고 가정한다.

- ① 현재 프레임에서의 타겟 위치를 앞 프레임에서의 위치인 \hat{y}_0 로 초기화한 후 $\{\hat{p}_u(\hat{y}_0)\}_{u=1, \dots, m}$ 를 계산하고,

$$\rho[\hat{p}(\hat{y}_0), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} \text{을 계산한다.}$$

- ② $w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[h(x_i) - u]$ 을 계산한다.

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_s} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_s} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}$$

- ③ 계산한다. (여기서, $g = -k'$)

- ④ $\{\hat{p}_u(\hat{y}_1)\}_{u=1, \dots, m}$ 을 계산하고, $\rho[\hat{p}(\hat{y}_1), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_1) \hat{q}_u}$ 을 계산한다.

- ⑤ $\rho[\hat{p}(\hat{y}_1), \hat{q}] < \rho[\hat{p}(\hat{y}_0), \hat{q}]$ 이면, $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_0 + \hat{y}_1)$ 로 세팅한 뒤 $\rho[\hat{p}(\hat{y}_1), \hat{q}]$ 계산한다. $\rho[\hat{p}(\hat{y}_1), \hat{q}] \geq \rho[\hat{p}(\hat{y}_0), \hat{q}]$ 이 만족될때까지 반복한다.

⑥ 만약 $\|\hat{y}_1 - \hat{y}_0\| < \epsilon$ 이면 그만하고, 아니면 $\hat{y}_0 \leftarrow \hat{y}_1$ 로 세팅하고 2번째 단계로 간다.

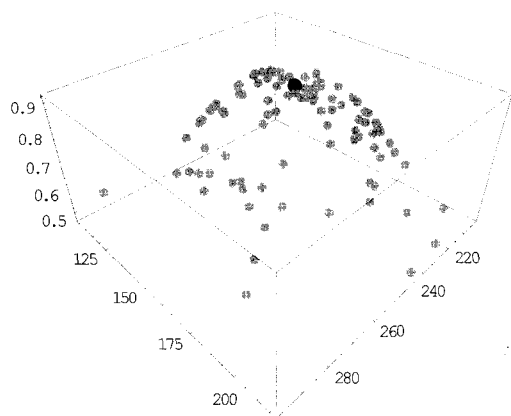
Mean-shift 알고리즘은 단일 후보를 이용하기 때문에 요구되는 계산 복잡도는 낮은 반면에, 주변의 지역 모드 위치로 수렴하게 되는 단점을 가지고 있다. 따라서, 겹침이 있거나 배경이 복잡한 환경에서는 좋은 성능을 발휘하지 못하게 된다.

4. Condensation 알고리즘과 Mean-shift 알고리즘을 결합한 추적방법

Condensation 알고리즘을 기반으로 하는 객체 추적방법은 많은 후보 개수로 인해 높은 계산 복잡도가 요구된다. 하지만, 후보들 중의 상당수가 비교적 낮은 유사도를 가지므로써 시스템의 효율을 떨어뜨린다. 따라서, 유사도가 높은 후보들만을 이용하는 것이 효율적이라고 할 수 있다. 이를 위해, 본 논문에서는 후보들의 유사도를 높이기 위해 그래디언트 방향과 동일한 방향을 가지는 Mean-shift 벡터 방향으로 후보의 위치를 바꿔주는 방법을 이용한다.

4.1 낮은 유사도를 가지는 후보들로 인한 비효율성

Condensation 알고리즘이 일반적으로 좋은 성능을 나타내지만, 많은 후보들을 이용한다. 하지만, 후보를 생성하는 단계에서 생성된 후보 집합 안에는 비교적 낮은 유사도(Likelihood)를 가지는 후보가 존재하는 경우가 많다. 예를 들어 (그림 1)은 단일 추적 실험에 쓰인 시퀀스의 한 프레임에서 Condensation 알고리즘을 이용한 객체 후보들의 위치와 유사도를 나타낸 것이다. 검은 색은 바람직한 객체의 위치와 그 때의 유사도를 나타내고, 회색 점들은 각 후보들의 위치와 유사도를 나타낸다. 상당수의 후보들이 낮은 유사도를 가지고 있는 것을 확인할 수 있다.



(그림 1) 객체 후보들의 유사도, 회색 점들은 각 후보들의 위치와 그 유사도를 나타내고, 검은색 점은 후보들에 의해 추정된 위치와 유사도를 나타낸다.

이러한 낮은 유사도를 가지는 후보들은 추정단계(estimation step)에서 객체 위치를 추정하는데 기여도가 낮기 때문에, 이러한 후보들이 많이 생성될수록 시스템의 효율성은 떨어지게 된다. 따라서, 많은 수의 후보를 생성해서 이용하기보다는 높은 유사도를 가지는 적은 수의 후보를 이용하는 것이 효율적이라고 볼 수 있다. 높은 유사도를 가지는 후보를 선택하기 위해서는 먼저 모든 후보들의 유사도를 계산한 후 그 중 높은 유사도를 가지는 후보들을 선택하는 방법이 있지만, 이 방법은 모든 후보에 대해서 Condensation 알고리즘의 관측 단계까지 모두 수행해야만 하기 때문에 매우 비효율적이라고 할 수 있다.

4.2 Mean-Shift를 이용한 후보 집합 생성

후보의 수가 적더라도 생성된 후보들이 높은 유사도를 가지는 것이 바람직하지만, 일반적으로 랜덤하게 생성된 후보들 중에는 낮은 유사도를 갖는 후보들이 다수 존재한다. Condensation 알고리즘은 후보 집합 안에서 낮은 유사도를 가지는 후보들의 수를 줄이기 위해 Factored Sampling 과정을 이용한다. 하지만, Factored Sampling 과정을 통해 비교적 높은 유사도를 가지는 후보들을 생성하기 위해서는 반드시 생성된 모든 후보들의 유사도를 계산해야만 하고, 객체 모델의 복잡도에 따라 많은 계산 복잡도가 요구되기도 한다.

이에 계산 복잡도를 줄이기 위해서는 객체 모델의 복잡도를 낮추거나 생성되는 후보들의 수를 줄여야 한다. 본 논문에서는 계산 복잡도를 줄이기 위해서 후보의 수를 줄이는 방법을 이용한다. 하지만, 후보의 수를 단순히 줄이는 것은 낮은 추적 성능을 가져오기 때문에 높은 유사도를 가지는 후보들을 생성해야만 한다.

이를 위해 본 논문에서는 Fukunaga와 Hostetler에 의해 제안된 Mean-shift 알고리즘을 이용한다. Mean-shift 알고리즘을 이용한 추적에서 후보의 초기 위치는 프레임의 실제 객체 위치와 가까운 위치이어야 하기 때문에, 후보들을 무작위로 생성하는 Condensation 알고리즘을 Mean-shift 알고리즘과 결합하는 방법만으로는 좋은 성능을 얻을 수 없다. 만일 실제 객체의 위치와 가까운 위치에서 후보가 생성되지 않는다면, Mean-shift 알고리즘을 이용해도 후보의 유사도를 높일 수 없기 때문이다. 이러한 후보의 생성 문제를 해결하기 위해 본 논문에서는 객체의 모션 벡터(Motion vector)를 이용한다.

객체의 모션 벡터를 구하기 위해 먼저 프레임이 그레이(Grey) 레벨을 갖는다고 가정한다. $I(x,y,t)$ 와 $I(x,y,t+1)$ 을 각각 $t, t+1$ 프레임의 이미지라고 할 때, 다음을 만족한다고 가정한다.

$$I(x + p(x,y,t), y + q(x,y,t), t + 1) = I(x, y, t) \quad (5)$$

여기서, $(p(x,y,t), q(x,y,t))$ 는 $t, t+1$ 프레임 사이의 객

체의 모션에 의한 이동(displacement)이다. 테일러 전개를 이용해 (5)식을 근사시키면 다음과 같다.

$$I(x+p, y+q, t+1) \cong I(x, y, t) + pI_x + qI_y + I_t \quad (6)$$

여기서 $I_x = \frac{\partial I(x, y, t)}{\partial x}$, $I_y = \frac{\partial I(x, y, t)}{\partial y}$, $I_t = \frac{\partial I(x, y, t)}{\partial t}$, $p = p(x, y, t)$, $q = q(x, y, t)$ 이다.

(5)식과 (6)식에 의해, 다음 조건을 만족한다[14].

$$pI_x + qI_y + I_t = 0 \quad (7)$$

따라서, 다음과 같은 에러함수를 최소로 하는 (p, q) 가 객체의 모션 정보이다.

$$Err^{(t)}(p, q) = \sum_{(x, y) \in R} (pI_x + qI_y + I_t)^2 \quad (8)$$

먼저, 후보를 선택하고, 생성하기 위해, Factored Sampling 기법을 이용해서, Sampling 된 후보 집합으로부터 다음과 같이 생성한다.

$$\mathbf{s}_t = A\mathbf{s}_{t-1} + \mathbf{w}_{t-1} \quad (9)$$

여기서, A는 model의 deterministic component이고, \mathbf{w}_{t-1} 은 multivariate random variable이다

다음으로, 생성된 후보들을 이용해서 각 후보에 대응되는 칼라 분포(Color Distribution)는 다음과 같이 표현된다.

$$P_y^{(n)} = f \sum_{i=1}^I k \left(\frac{\|y^n - x_i^n\|}{a} \right) \delta[h(x_i^n) - u] \quad (10)$$

여기서, n은 후보 인덱스(Sample Index), I는 타원 안의 pixel의 수, δ 는 Kronecker delta 함수, a는 $\sqrt{H_x^2 + H_y^2}$, f는 normalization factor이다.

생성된 후보들의 칼라 분포를 이용해서, 다음의 weight를 다음과 같이 계산한다.

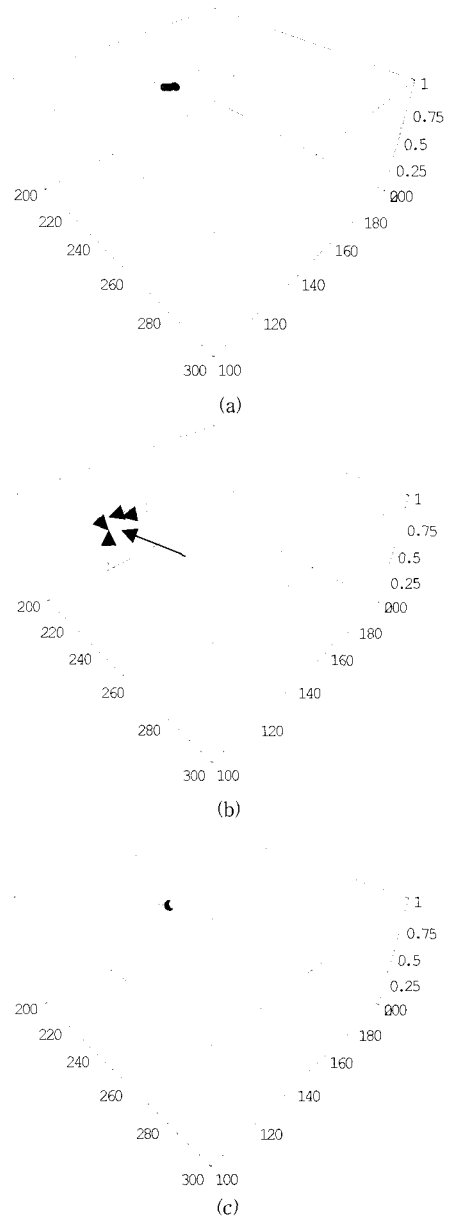
$$w_i^n = \sum_{u=1}^m \sqrt{\frac{\hat{q}}{\hat{p}_n(y_0)}} \delta[h(x_i^n) - u] \quad (11)$$

Mean-shift 알고리즘과 계산된 weight들을 이용해서, Mean-shift 벡터를 다음과 같이 계산한다.

$$\mathbf{y}^n = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i^n w_i^n g\left(\frac{\|\mathbf{y}_0^n - \mathbf{x}_i^n\|}{a}\right)}{\sum_{i=1}^{n_h} w_i^n g\left(\frac{\|\mathbf{y}_0^n - \mathbf{x}_i^n\|}{a}\right)} - \mathbf{y}_0^n \quad (12)$$

Mean-shift vector는 항상 그레디언트(gradient) 방향을 가리키므로, 다음과 같이 계산되어지는 위치에서의 후보들의 유사도는 항상 다음을 만족하게 된다.

$$\rho[\mathbf{p}(y_0^n), \mathbf{q}] \leq \rho[\mathbf{p}(y^n), \mathbf{q}] \quad \text{where } \mathbf{y}_i^n = \mathbf{y}_0^n + \mathbf{y}^n \quad (13)$$



(그림 2) Mean-Shift를 이용한 후보의 이동 : (a) 생성된 후보의 위치와 유사도 (b) Mean-Shift 알고리즘을 이용한 후보의 이동 (c) 이동된 후보들의 위치와 유사도

즉, Mean-shift 벡터 방향으로 이동된 후보의 유사도는 항상 그 전에 비해 상대적으로 높아지게 된다.

따라서, 이동된 후보들을 새로운 후보 집합으로 하면, 보다 정확한 추정을 할 수 있게 된다. (그림 2)는 이러한 Mean-Shift 알고리즘에 의한 후보의 위치 변화를 도식화 한 것이다.

(그림 2) (a)는 생성된 후보의 위치와 유사도를 나타낸 것이다. 생성된 후보는 비교적 낮은 유사도를 나타내고 있다. (그림 2) (b)는 생성된 후보의 위치를 Mean-Shift 벡터 방향으로 이동하는 것을 나타낸 것이다. (b)에서 볼 수 있듯이, 각각의 후보들은 유사도가 높아지는 방향으로 이동되는 것을 볼 수 있다. (c)는 이동 후의 후보들의 위치와 유사도를 나타낸 것이다. (c)에서 볼 수 있듯이, 모든 후보들이 유사도가 높은 곳에 위치하고 있음을 확인할 수 있다.

(그림 3)은 각 추적방법을 이용해서 객체의 위치를 추정했을 때와 제안한 Condensation 알고리즘과 Mean-shift 알고리즘을 이용해서 객체를 위치를 추정했을 때, 그리고 실험

제 객체의 위치와 유사도를 나타낸 것이다. (그림 3) (a)에서 볼 수 있듯이, 각 추적방법에 의해 추정된 위치에서의 유사도를 보면 비교적 높은 유사도를 나타내는 것을 알 수 있다. 하지만 (그림 3) (b)에서 볼 수 있듯이, Condensation 추적방법의 경우, 낮은 유사도를 가지는 후보들의 영향으로 인해 비교적 낮은 성능을 나타내고 있고, 또한 많은 수의 후보를 이용함으로써, 낮은 효율성을 가지고 Mean-Shift 추적방법의 경우에도, 단일 후보를 가지는 특성과 로컬 모드로 수렴하는 특성으로 인해 비교적 낮은 성능을 나타내고 있다.

하지만, 이에 비해, 제안된 추적방법의 경우에는, 생성한 후보들의 위치를 Mean-Shift 방향으로 이동시켜 후보들의 유사도를 높임으로서 적은 수의 후보로도 비교적 좋은 결과를 나타내고 있다.

5. 실험 및 결과

5.1 실험 환경

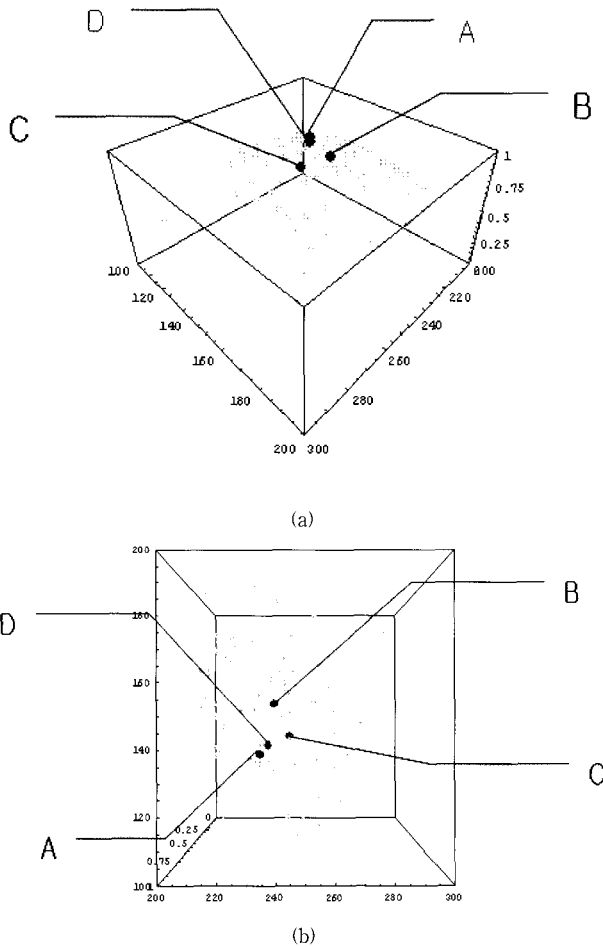
본 실험은 CPU P4-1.5GHz와 메모리 512MB의 성능을 가지는 PC에서 수행했다. OS는 Microsoft사의 Windows XP를 이용했고, 개발 툴은 Microsoft Visual C++와 Intel사의 IPL(Image Processing Library), OpenCV(Open Computer Vision Library)를 이용했다. 모든 시퀀스는 320×240 해상도를 가지는 비트맵 시퀀스(Bitmap Sequence)를 사용했다.

K. Nummiaro[12-13]가 제안한 Condensation 추적기는 100-150개의 후보를 사용했다. 본 논문에서는 Condensation 추적 방법과의 비교를 위해 이와 비슷한 100개의 후보를 이용하는 Condensation 추적 시스템을 구성했다. 또한 Mean-shift를 이용한 추적 시스템을 구성하는데는 D. Comaniciu [3-5]가 제안한 방법을 이용했다. 그리고 제안된 방법을 이용한 추적 시스템의 경우에는 후보 5개를 이용했다.

단일 추적 실험에 쓰인 시퀀스는 사람이 계단을 오르는 시퀀스와 사람이 나무에 가려지는 시퀀스, 그리고 사람이 책장 앞에서 움직이는 시퀀스이고, 다중 객체 추적 실험에 쓰인 시퀀스는 두 사람의 얼굴이 겹쳐지는 시퀀스이다.

5.2 단일 추적(Single-Tracking)

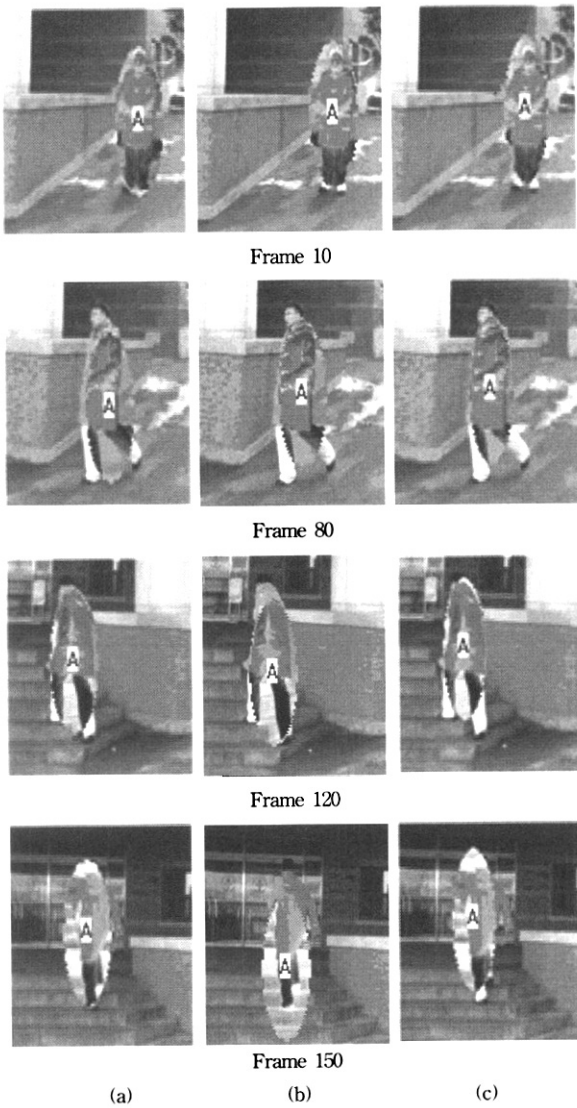
(그림 4)는 Condensation 추적기, Mean-shift 추적기, 제안된 추적기의 단일 추적(Single Tracking) 결과를 나타낸 것이다. (그림 4) (a)를 보면, 100개의 후보를 이용한 Condensation 추적기의 경우, 좋은 추적 결과를 얻을 수 있었지만, 반면, 많은 후보를 이용함으로써, 높은 계산 복잡도가 요구됨을 알 수 있다. <표 1>과 같이, Mean-shift 추적기의 경우에는 단일 후보를 이용하기 때문에 낮은 계산 복잡도로 인해 프레임당 처리시간이 적게 소요되지만, 그림 4 (b)에서의 150 프레임에서와 같이 지역 모드(Local mode)로 수렴하는 특성으로 인해 상대적으로 낮은 성능을 나타내고 있다. 이에 비해 제안된 추적기의 경우, 적은 수의 후보를 이용함



(그림 3) 각 추적 방법에 의한 객체 위치 추정 : (a) 각 추적방법에 의해 추정된 위치와 그 유사도 (b) 이미지 좌표 평면 상에서의 각 추적방법에 의해 추정된 위치(A. 실제 위치, B. Condensation 알고리즘에 의해 추정된 위치 C. Mean-Shift에 의해 추정된 위치 D. 제안된 방법에 의해 추정된 위치)

으로서, Condensation 추적기에 비해 상대적으로 낮은 계산 복잡도를 가지기 때문에 Condensation 추적기에 비해 적은 프레임당 처리 시간이 들지만, (그림 5) (c)와 같이 비교적 좋은 추적 결과를 보이고 있음을 볼 수 있다. 이는, 비록 적은 수의 후보라도, 유사도가 높은 후보를 이용했기 때문이라고 볼 수 있다. (그림 5)는 (그림 4)에서 보인 단일 추적 경우의 Condensation 추적기와 제안된 추적기에서 사용한 후보들의 평균 유사도를 나타낸 것이다. (그림 5)에서 볼 수 있듯이, Mean-shift 추적기의 경우, 단일 후보가 지역 모드 위치로 수렴하는 특성으로 인해 후보의 유사도가 낮게 나오지만, 제안된 추적기의 경우에는 5개의 후보를 이용하더라도 사용되는 후보들의 평균 유사도가 100개의 후보를 사용할 때보다도 높게 나오는 것을 볼 수 있다.

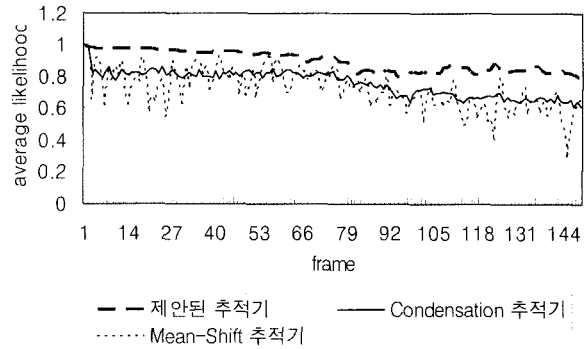
(그림 6)은 각 추적기의 오차를 나타낸 것이다. (그림 6)에서 Mean-shift 추적기와 제안된 추적기가 128 frame 근처



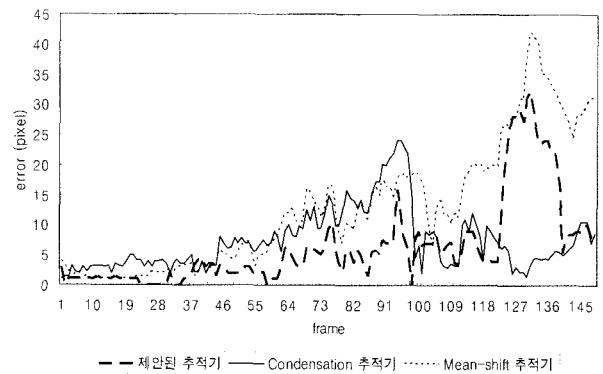
(그림 4) 각 추적기의 단일 추적 결과: Condensation 추적기 (a) Mean-shift 추적기 (b) 제안된 추적기 (c)

〈표 1〉 단일 추적의 경우 프레임당 처리 시간

	Condensation	Mean-shift	제안된 추적기
프레임당 처리시간 (sec)	0.084	0.007	0.035



(그림 5) 단일 추적 경우 이용된 후보들의 평균 유사도

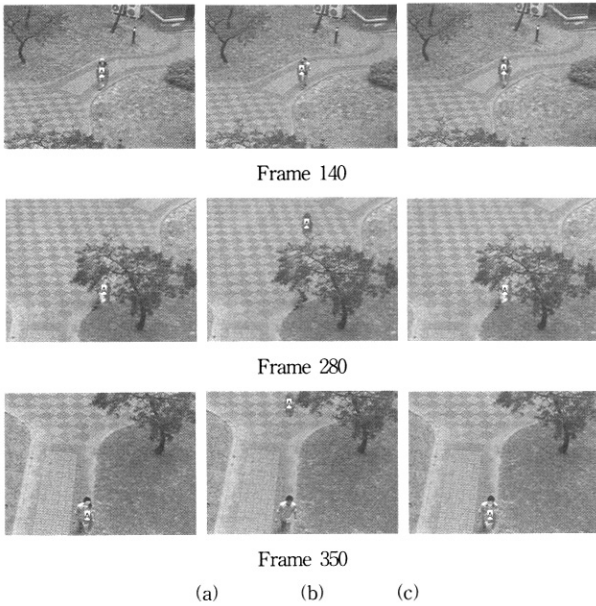


(그림 6) 단일 추적에서의 각 추적기의 오차

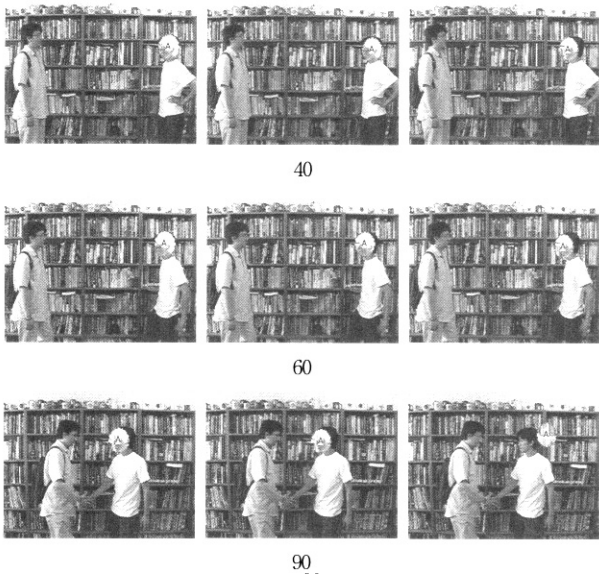
에서 갑자기 오차가 급증하는 것을 볼 수 있다, 이것은 Mean-shift가 전역 모드(global mode) 위치가 아닌 지역 모드(local mode) 위치로 수렴하는 특성 때문에 나타난다. 하지만, 제안된 추적기의 경우, 단일 후보를 사용하기 때문에, 잘못된 위치로 수렴을 하면 추적에 실패하는 Mean-shift 추적기와는 달리, 복수개의 후보를 이용하기 때문에 후보들이 잘못된 위치로 수렴하더라도, 다시 추적에 성공하게 된다.

(그림 7)은 장애물에 의해 객체에 겹침이 발생하는 환경에서의 각 추적기의 결과를 보인 것이다. 100개의 후보를 사용하는 Condensation 추적기와 5개의 후보를 사용한 제안된 추적기의 경우에는 비록 객체에 겹침이 발생하더라도 추적에 성공하는데 반면, 단일 후보를 이용하는 mean-shift 추적기의 경우에는 겹침이 발생하면 추적에 실패한다.

(그림 8)은 복잡한 배경에서의 각 추적기의 결과를 보인 것이다. 단일 후보를 사용하는 Mean-shift 추적기의 경우, 복잡한 배경에 의한 지역 모드 위치로 수렴해 추적에 실패하지만, 복수개의 후보를 이용하는 Condensation 추적기와 제안된 추적기의 경우에는 추적에 성공하게 된다.



(그림 7) 장애물에 의해 객체에 겹침이 발생하는 환경에서의 추적결과(a) Condensation 추적기 (b) Mean-shift 추적기 (c) 제안된 추적기



(그림 8) 복잡한 배경 환경에서의 추적결과 : (a) Condensation 추적기 (b) Mean-shift 추적기 (c) 제안된 추적기

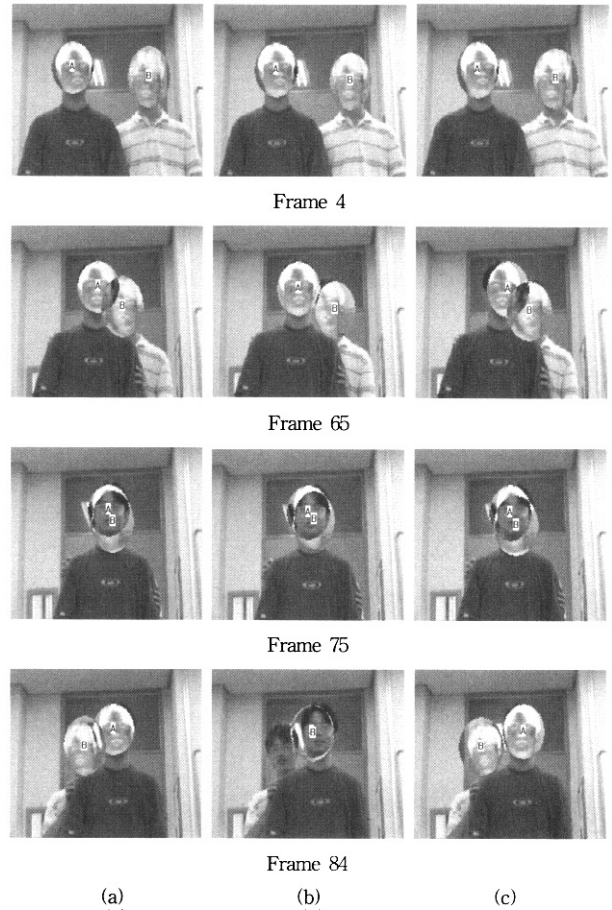
5.3 다중 추적(Multiple Tracking)

(그림 9)은 Condensation 추적기, Mean-shift 추적기, 제안된 추적기의 다중 추적(Multiple Tracking) 결과를 나타낸 것이다. (그림 9)을 보면 Condensation 추적기 경우와 제안된 추적기의 경우에는 겹침(Occlusion)이 발생하더라도, 비교적 좋은 추적 결과를 보이지만, Mean-shift 추적기의 경우 겹침이 발생 후에는 추적에 실패하게 된다. 이는 앞에서 언급한 Mean-shift 추적기의 수렴성 때문이다. 제안된 추적기는 복수개의 후보를 사용하기 때문에, 이러한 문제가 발

생하지 않고, 또한 후보의 개수도 상대적으로 Condensation 추적기에 비해 적기 때문에 효율적이라고 할 수 있다.

<표 2>는 다중 추적 경우의 각 추적기 프레임 당 처리 시간을 나타낸 것이다. 단일 추적일 때와 마찬가지로 Condensation 추적기에 비해 제안된 추적기의 프레임 당 처리 시간이 적게 소요되는 것을 확인할 수 있다.

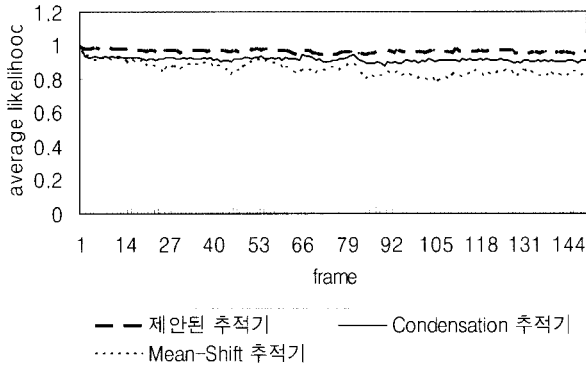
(그림 10)은 (그림 9)에서 보인 다중 추적 경우의 각 추적기와 제안된 추적기에서 사용한 후보들의 평균 유사도를 나타낸 것이다. 단일 추적 경우와 마찬가지로, (그림 10)에서도 후보들의 평균 유사도를 보면, 제안된 추적기의 경우가 가장 높게 나타나는 것을 볼 수 있다. (그림 10)에서 볼 수 있듯이, 단일 후보를 이용하는 Mean-shift 추적기나 100개의 후보를 이용하는 Condensation 추적기보다도 제안된 추



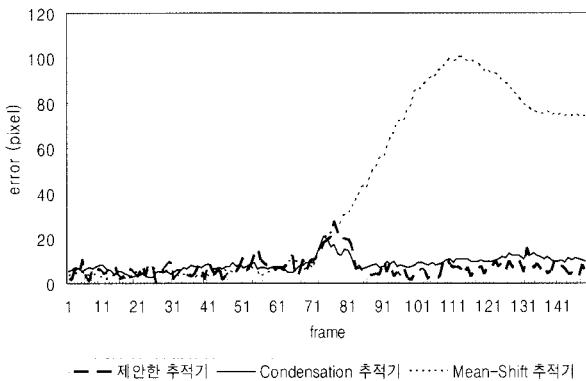
(그림 9) Condensation 추적기와 제안된 추적기의 다중 추적 결과 : Condensation 추적기 (b) Mean-shift 추적기 (c) 제안된 추적기

<표 2> 다중 추적 경우 프레임 당 처리 시간

	Condensation	Mean-shift	제안된 추적기
프레임당 처리시간 (sec)	0.248	0.017	0.086



(그림 10) 다중 추적 경우 이용된 후보들의 평균 유사도



(그림 11) 다중 추적에서의 각 추적기의 오차

적기에서 사용하는 후보들의 평균 유사도가 높다는 것을 볼 수 있다.

(그림 11)은 다중 추적에서의 각 추적기의 오차를 나타낸 것이다. Mean-Shift 추적기의 경우 겹침으로 인해 추적에 실패하게 되고, Condensation 추적기보다는 전반적으로 적은 오차를 나타내고 있는 것을 확인할 수 있다.

6. 결 론

실시간 추적(Real-Time) 추적은 객체 추적에서 매우 중요한 문제이다. 본 논문에서는 Mean-Shift 알고리즘을 이용해 Condensation 알고리즘을 이용해 추적을 할 경우, 적은 수의 후보를 이용해서 추적을 하는 모델을 제안했다.

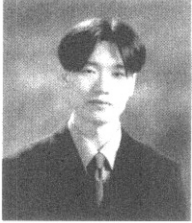
제안된 모델의 경우, 적은 후보를 이용하지만, 높은 유사도를 가지는 후보를 이용함으로써, Condensation 알고리즘보다 더 효율적인 추적을 보여 주었고, 또한 복수개의 후보를 이용하기 때문에, 단일 후보를 이용함으로써 생기는 Mean-shift 알고리즘의 단점도 보완할 수 있었다.

하지만 제안된 시스템의 경우, 시스템의 성능을 좌우하는 여러 파라미터들을 적절히 바꿔줘야 하는 단점을 가지고 있다. 이러한 시스템 파라미터들을 적응적으로 적용할 수 있는 방식을 향후 연구되는 것이 필요하다.

참 고 문 헌

- [1] M. Isard, A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density", European Conference on Computer Vision, pp.343-356, 1998.
- [2] M. Isard, A. Blake, "Condensation - Conditional Density Propagation for Visual Tracking", International Journal on Computer Vision Vol.1, No.29, pp.5-28, 1998.
- [3] D. Comaniciu and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean shift", proc. IEEE Conf. Computer Vision and pattern Recognition, vol 2, pp. 142-149, June, 2000.
- [4] D. Comaniciu and P. Meer, "Mean shift : A Robust Approach Toward Feature Space Analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.24, No.5, pp.603-619, May, 2002.
- [5] D. Comaniciu and V. Ramesh, P. Meer, "Kernel-Based Tracking", IEEE trans. Pattern Analysis and Machine Intelligence, Vol.25, No.5, pp.564-577, May, 2003.
- [6] R.E.Kalman "A New Approach to Linear Filtering and Prediction Problems" Transactions of the ASME, Series D, Journal of Basic Engineering, Vol.82 No.1, pp.34-45, 1960.
- [7] Y.Boykov and D.Huttenlocher, "Adaptive Bayesian Recognition in Tracking Rigid Objects", Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.697-704, 2000.
- [8] R. Rosales and S. Sclaroff, "3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.117-123, 1999.
- [9] K. Fukunaga and D. Hostetler "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition" IEEE trans. Information Theory Vol.IT-21, pp.32-40, 1975.
- [10] R. Collins, "Mean-shift Blob Tracking through Scale space", IEEE Computer Society Conference on CVPR'03, 2003.
- [11] P. Perez, C. Hue, J. Vermak, M Gangnet, "Color-Based Probabilistic Tracking, European Conference on Computer Vision, pp.661-675, 2002.
- [12] K. Nummiaro, E. Koller-Meier and L. Van Gool, "A Color-Based Particle Filter", First International Workshop on Generative-Model-Based Vision, in Conjunction with ECCV'02, pp.53-60, 2002.
- [13] K. Nummiaro, E. Koller-Meier and L. Van Gool, "Object Tracking with an Adaptive Color-Based Particle Filter", Symposium for Pattern Recognition of the DAGM, pp. 353-360, 2002.

[14] B.K.P. Horn and B.G. Schunck. Determining optical flow. Artificial Intelligence, 185-203, 1981.



조 상 현

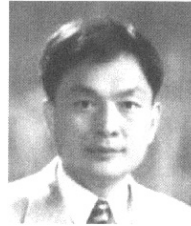
e-mail : cshgreat@catholic.ac.kr

2003년 가톨릭대학교 자연과학부(이학사/공학사)

2005년 가톨릭대학교 대학원 컴퓨터 공학과(공학석사)

2005년~현재 가톨릭대학교 대학원 컴퓨터 공학과 박사과정

관심분야: 영상처리, 컴퓨터 비전, 패턴인식 등



강 행 봉

e-mail : hbkang@catholic.ac.kr

1980년 한양대학교 전자공학과 졸업

1986년 한양대학교 대학원 전자공학과(석사)

1989년 미국 Ohio State University 컴퓨터 공학 석사

1993년 미국 Rensselaer Polytechnic Institute 컴퓨터 공학 박사

1994년~1997년 삼성종합기술원 수석연구원

1997년~현재 가톨릭대학교 컴퓨터정보공학부 부교수

관심분야: 컴퓨터 비전, 멀티미디어 시스템, 인공 지능, 생체 인식 및 Bioinformatics