

움직임 카메라 환경에서 파티클 필터를 이용한 객체 추적

정회원 고병철*, 남재열*, 준회원 곽준영*

Object Tracking Using Particle Filters in Moving Camera

Byoung-Chul Ko*, Jae-Yeal Nam* *Regular Members*, Joon Young Kwak* *Associate Member*

요약

본 연구에서는 움직이는 CCD 카메라로부터 입력된 영상에서 색상 및 질감 성분을 기반으로 하는 파티클 필터를 이용하여 실시간으로 객체를 추적할 수 있는 알고리즘을 제안한다. 초기 영상에서 추적하고자 하는 객체를 선택하면 이를 타깃 파티클로 결정하고, 타깃 파티클로 부터 추적을 위한 초기 상태가 모델링 된다. 이후 프레임 부터 N개의 파티클들이 랜덤 분포로 생성되고 각 파티클로 부터 질감 정보인 로컬 CS-LBP (Centre Symmetric Local Binary Patterns) 모델과 색상 분포 모델이 특징 모델로 사용된다. 각 특징 모델에 대해 바타차리야 (Bhattacharyya) 거리를 사용하여 각 파티클과 타깃 파티클 간의 특징 관측 우도(likelihood)를 구하고 이를 각 파티클의 가중치로 설정 한다. 각 파티클의 가중치를 기반으로 가중치가 가장 높은 파티클을 새로운 타깃으로 설정 하고, 각 파티클들을 재 샘플링 한다. 본 실험결과에서는 여러 가지 특징을 조합하여 실험을 하였고, 그 결과 색상 분포 모델과 로컬 CS-LBP를 조합했을 때 추적 성능이 가장 우수한 것을 확인할 수 있었다.

Key Words : 객체 추적, 색상 분포 모델, 로컬 CS-LBP, 관측 우도, 바타차리야 거리

ABSTRACT

This paper proposes a new real-time object tracking algorithm using particle filters with color and texture features in moving CCD camera images. If the user selects an initial object, this region is declared as a target particle and an initial state is modeled. Then, N particles are generated based on random distribution and CS-LBP (Centre Symmetric Local Binary Patterns) for texture model and weighted color distribution is modeled from each particle. For observation likelihoods estimation, Bhattacharyya distance between particles and their feature models are calculated and this observation likelihoods are used for weights of individual particles. After weights estimation, a new particle which has the maximum weight is selected and new particles are re-sampled using the maximum particle. For performance comparison, we tested a few combinations of features and particle filters. The proposed algorithm showed best object tracking performance when we used color and texture model simultaneously for likelihood estimation.

I. 서론

최근 들어 비디오 기반 감시 시스템에 요구가 증가 하면서 감시 시스템에서 가장 중요한 요소인 객체 추적에 대한 연구가 활발히 진행되고 있다. 객체 추적 기술은 감시 시스템이외에도 행동 인식, 모바

일 로봇, 스마트 룸, 증강현실, 비디오 압축 등과 같이 컴퓨터 비전 분야에서 중요한 역할을 한다. 하지만 현실에서 실시간 객체 추적은 객체의 가려짐, 낮은 해상도, 조명의 변화, 배경의 복잡함 등으로 인해 정확한 추적이 힘든 문제점이 있다. 이러한 문제점을 해결하기 위해서 최근 다양한 연구가 활발

※ 본 논문은 연구재단의 '대구과학연구단지육성사업 융합기술 기초 R&D 지원사업(2011)'의 연구기금으로 수행 되었음.

* 계명대학교 컴퓨터공학과 CVPR 연구실(miceko@kmu.ac.kr) (° : 교신저자)

논문번호 : KICS2012-02-056, 접수일자 : 2012년 2월 9일, 최종논문접수일자 : 2012년 4월 27일

히 진행되고 있다.

객체 추적은 크게 두 가지 방향으로 연구가 진행되고 있다^[1]. 첫 번째로 결정적 방법 (deterministic method)은 전형적으로 템플릿 영상과 현재 영상 사이에 유사도 비용 함수의 지역적 극대점을 반복적으로 찾는 방법으로 추적을 수행한다. 템플릿 영상과 현재 영상 사이의 비용 함수로는 최소제곱오차합(SSD: Sum of Squared Differences)이 주로 사용되지만, mean-shift 알고리즘^[2]이 최적 해를 찾기 위한 강건한 유사도 측정 방법으로 알려져 있다.

두 번째로 통계적 방법(stochastic method)은 추적 시스템의 다양성을 모델링하기 위해 상태 공간(state space)을 이용한다. 연속적 몬테카를로(Sequential Monte Carlo)로 알려진 파티클 필터가 대표적인 통계적 방법이다. 몬테카를로법은, 물리적, 수학적 시스템의 행동을 시뮬레이션하기 위한 계산 알고리즘이며 다른 알고리즘과는 달리 통계학적이고, 무작위의 숫자를 사용하는 비결정적인 방법이다^[3]. 파티클 필터 방법은 몬테카를로법을 이용하여 반복적으로 상태 공간의 사후 확률을 측정하고 이를 기반으로 객체를 추적한다. 최근에는 대부분의 연구가 파티클 필터를 기반으로 하는 통계적 방법을 이용함으로 본 논문에서도 파티클 필터를 이용하여 객체를 추적하는 연구에 초점을 맞추고 관련 연구를 소개한다.

Yang^[1]등은 파티클 필터에 계층적 (Cascade) 모델을 적용한 방법을 제안 하였다. 첫 번째 단계에서 컬러 특징과 HOG (Histogram of oriented gradient) 특징을 사용하고, 두 번째 단계에서 Haar-like 특징을 사용한다. 마지막 단계로 SIFT(Scale Invariant Feature Transform) 기술자를 사용하여 최종적으로 타깃 파티클을 구한다.

Deguchi^[4] 등은 객체를 추적하기 위해 mean-shift 알고리즘을 사용하여 객체의 위치를 추정하고 객체의 가려짐이나 추적시간을 줄이기 위해 파티클 필터를 병행적으로 사용하는 방법을 제안하였다.

Islam^[5]등은 파티클 필터와 Distance transformation (DT)을 사용하여 객체를 추적하는 방법을 제안 하였다. 객체의 변화에 덜 민감한 DT 이미지를 생성하였고, 유사도 측정을 위해 크기와 회전에 민감하지 하도록 정규화한 교차상관관계 (cross-correlation) 방법을 제안 하였다.

Klein^[6]등은 각 파티클 윈도우에서 윈도우 영역을 지역(local)으로 분할하고 각 지역에서 Haar-like 특징을 추출한다. 관측 우도(likelihood)를 계산하기

위해 Ada-Boost방법을 응용하여 각 특징들에 대해 약 분류기를 매 프레임마다 학습시키고 이를 기반으로 강한 분류기를 생성하여 객체를 추적하는데 사용하였다.

Sidibe^[7]등은 파티클 필터에 시각적 중요 정보를 결합하여 객체 간의 겹침이 발생하거나 조명 변화가 심하게 발생하는 경우에도 중요 객체를 추적할 수 있는 알고리즘을 제안하였다.

Khan^[8]등은 파티클 필터와 멀티 모달 비등방성 mean-shift를 이용하여 객체를 추적하는 방법을 제안하고 있다. 이 논문에서는 추적을 위해 객체의 다양한 모양과 외형을 추정하고 참조 객체에 대해 온라인 학습을 실시한다. 또한, 멀티 모달 mean-shift를 이용하여 객체의 외형 박스 안에 포함된 5가지 파라미터(중심점 (cx, cy), 폭, 높이, 기울기 각도)를 조절하여 가변적으로 추적 객체에 적합한 파티클을 결정하는 방법을 사용하고 있다.

Chau^[9]등은 다양한 특징들에 대한 유사성 측정 방법과 궤도(trajecory) 필터링 방법에 의해 객체를 추적하는 방법을 제안하고 있다. 거리, 영역, 모양 비율, 색상 히스토그램의 특징 집합을 각 추적 객체로부터 추출하고 최적의 매칭 객체와 객체의 상태를 예측하기 위해 Kalman 필터를 사용하고 있다. 또한, 객체의 가림이나 오검출로 인해 잘못된 객체를 추적하는 것을 방지하기 위해 궤도 필터링 방법을 제안하여 추적의 성능을 높이고 있다.

표 1은 지금까지 설명한 여러 가지 추적 알고리즘을 특징 벡터와 추적 방법으로 분류한 것이다. 표 1에서 살펴본 바와 같이 기존의 연구에서는 파티클 필터, mean-shift와 함께 색상 히스토그램^[6,9]이나 색상 분포 모델^[4,7], HOG (Histogram of oriented gradient)[1], 모양^[8,9] 등을 주요 특징으로 사용하고 있는 것을 알 수 있다.

기존의 방법에서 색상 정보는 잡음, 객체의 기울어짐, 부분적 가림 등에 강건하고 계산 속도가 빠른 장점이 있지만^[10], 조명의 변화에 민감하고 객체의 겹침이 심할 경우 추적 성능이 떨어지는 단점이 있다. 이를 개선하기 위해 영상 내에 포함된 객체의 픽셀 그래디언트의 방향을 이용하는 HOG가 사용되기는 하지만 특징 벡터의 크기가 한 파티클 당 3780차원이 추출됨으로 후보 파티클 들에 대해 모두 적용하기 위해서는 계산 시간이 많이 걸리는 단점이 있다. 또한, 기존 연구들은 고정된 카메라를 기반으로 연구되었으므로 바람, 지반의 흔들림, 모바일 로봇의 움직임 등으로 인해 카메라에서 떨림

이 발생할 경우 객체를 놓치는 경우가 발생한다.

본 논문에서는 잡음 및 객체의 기울어짐, 부분적 가림, 카메라의 떨림에 강건한 알고리즘을 개발하기 위해 우선 색상 분포에 대해 지역적인 픽셀의 위치를 고려하여 가중치 색상 모델을 생성하였다. 또한, 객체의 겹침, 조명 변화에 강인하도록 추적 파티클을 4x4 크기의 지역적 영역으로 분할하고 각 영역에서 추출된 질감 정보인 CS-LBP (Centre Symmetric Local Binary Patterns)에 대한 특징 모델을 각각 생성하여 사용하였다. 이렇게 생성된 특징 모델은 관측모델로 사용되어 타깃과 후보 파티클들의 관측 우도를 사용하는데 사용된다. 각 파티클들의 최종 가중치를 측정하기 위해, 두 가지 관측 우도 결과를 선형 가중 결합하여 조명과 객체 겹침 등에 덜 민감한 파티클 가중치를 생성할 수 있다. 새로운 타깃의 상태를 결정하기 위해 최대값을 갖는 가중치를 선택하는 대신에 상위 N개의 상태 벡터를 조합하여 최종 타깃의 상태를 결정하도록 알고리즘을 제안하였다. 또한, 카메라의 움직임과 추적 객체의 크기 변화를 고려하여 객체를 추적할 수 있도록 바타차리야 (Bhattacharyya) 거리와 1차 자기회귀 (autoregressive)를 적용하여 파티클 필터를 예측하는 알고리즘을 제안하고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 객체 색상 분포 모델과 질감 분포 모델 생성 방법을 설명한다. 3장에서는 파티클 필터와 자기회귀 모델을 이용하여 객체를 추적하는 알고리즘을 설명한다. 4장에서는 제안된 알고리즘과 기존의 알고리즘간의 성능을 비교분석하고 5장에서는 본 논문에 대한 결론을 서술한다.

II. 관측 모델 (observation model) 생성

2.1 객체 색상 분포 모델 생성

객체 추적을 위해 사용되는 여러 가지 특징 중에서 색상 성분은 잡음이나 부분적인 가려짐에 강하고 영상의 회전이나 크기 변화에 덜 민감하며 계산 속도가 빨라 객체 추적을 위해 매우 중요한 정보가 된다^[8]. 따라서 본 논문에서는 우선적으로 색상 정보를 추적을 위한 특징 값으로 사용 하였다. 일반적으로 RGB색상은 HSV 색상에 비해 조명 변화에 더 민감함으로 파티클로 부터 HSV 색상을 추출하여 색상 분포 모델을 생성한다.

우선 타깃 파티클로 부터 m개의 빈을 갖는 색상 히스토그램 모델을 생성하고 이를 $p(x) = \{p_u(x)\}_{u=1...m}$ 으로 표기한다. p_u 는 u-번째 빈의 요소를 의미한다. 하지만 일반적으로 객체 파티클에서 주변보다는 중심 영역이 더 중요한 정보를 포함하고 있음으로, 타깃 파티클의 중심에 해당하는 픽셀에 더 많은 가중치를 주기 위해 x를 중심으로 하는 파티클의 색상 분포 모델을 식 (1)을 이용하여 갱신한다.

$$p_u(x) = c \sum_{x \in R} K \left(\left\| \frac{x_i - x}{h} \right\|^2 \right) \delta [b(x_i - u)] \quad (1)$$

식 (1) 에서 $b(x_i)$ 는 x_i 좌표에서의 색상 요소를 의미하고, δ 는 Kronecker 함수를, R은 파티클 영역을 의미 하며 c는 색상 분포를 0~1로 정규화 하기 위한 상수를 나타낸다. 함수 K는 Epanechnikov 커널을 의미하고 h는 파티클 영역의 반경을 나타낸다. 식(1)에 의해 색상 분포 모델은 파티클의 중심에 속한 픽셀의 경우 높은 가중치를 갖게 되고 주변으로 갈수록 낮은 가중치를 갖게 되어 물체에 의한 객체의 가려짐에 영향을 덜 받을 수 있게 된다.

표 1. 주요 추적 알고리즘의 특징벡터와 추적 방법
Table 1. The feature vector of main tracking algorithms and tracking methods

주요 알고리즘	특징 벡터	추적 방법
Yang ^[1]	색상 히스토그램, HOG, Haar-like특징, SIFT	particle filter, cascade model
Deguchi ^[4]	색상 분포 모델	mean-shift, particle filter
Islam ^[5]	Distance Transformation	particle filter, cross-correlation
Klen ^[6]	Haar-like특징	ada-boost
Sidibe ^[7]	색상 분포 모델, saliency 특징 정보	particle filter
Khan ^[8]	모양, 외형 정보	particle filter, multi-modal mean-shift
Chau ^[9]	거리, 영역, 모양 비율, 색상 히스토그램	Kalman-filter, trajectory filter

HSV의 각 채널로부터 0~1의 값을 갖는 128차원의 색상 분포 모델이 생성되어 한 파티클로부터 추출되는 총 특징 차원은 $128 \times 3 = 384$ 차원이 된다.

타깃 객체에 대한 색상 분포 모델이 결정되면 추적 대상이 되는 y 좌표 후보 파티클에서도 동일한 색상 모델 $p(y) = \{p_u(y)\}_{u=1\dots m}$ 을 생성한다.

2.2. 로컬 CS-LBP를 이용한 질감 분포 모델 생성

최근 객체의 검출과 추적을 위해 많이 사용되고 있는 질감 모델인 LBP (Local Binary Patterns) 및 HOG 특징들은 우수한 성능에 비해 특징의 차원수가 커서 실시간 객체 추적을 불가능 하게 하는 요인이 되고 있다. 본 논문에서는 이러한 문제점을 해결하기 위해 LBP와 비슷한 성능을 가지나 더 적은 특징 차원을 갖는 질감 특징으로서 CS-LBP (Center-symmetric LBP)를 사용한다. CS-LBP는 LBP와 비슷한 성능을 나타내고 있지만, 더 적은 특징 차원을 사용한다는 이점이 있다^[11].

CS-LBP는 중심 픽셀의 이웃하는 픽셀들의 관계를 나타내는 일종의 연산자이다. 중심 픽셀과 거리가 1인 이웃하는 4개의 픽셀들을 식 (2)에 적용하고, 그 결과를 반시계 방향으로 정렬하여 중심 픽셀에 대한 CS-LBP값을 얻는다. LBP의 경우 총 256 차원이 필요하지만, CS-LBP는 동일한 성능을 보이면서 총 16차원으로 차원을 줄일 수 있는 장점이 있다.

$$CS-LBP_{R,N}(x,y) = \sum_{i=0}^{(N/2)-2} s(n_i - n_{i+(N/2)})2^i \quad (2)$$

$$s(x) = \begin{cases} 1 & \text{if } x \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

식 (2)에서 n_i 와 $n_{i+(N/2)}$ 는 n_c 를 중심으로 반경 R 인 원 안에서 N 만큼씩 이동하며 대칭되는 점의 좌표를 의미한다. 그림 1에서 LBP의 경우 n_c 를 중심으로 중심과 각 점 n_i 와의 거리 값이 식 (3)의 조건을 만족 여부에 따라 1 또는 0의 값을 갖고 이를 반시계 방향으로 나열하여 8비트의 코드 값을 생성하지만, CS-LBP의 경우는 n_c 를 중심으로 대칭되는 점만의 거리 값을 계산하고 식 (3)의 조건 만족 여부에 따라 1 또는 0의 값을 반시계 방향으로 나열하여 4비트의 코드 값을 생성한다.

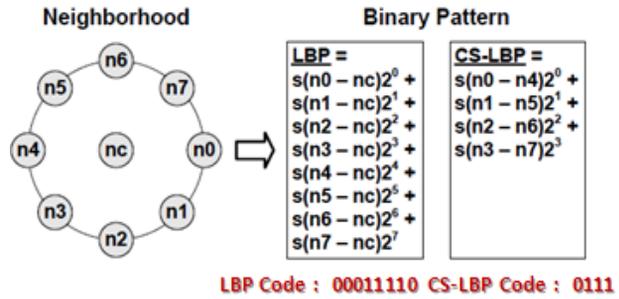


그림 1. LBP와 CS-LBP 특징 추출 방법
Fig. 1. Method of LBP and CS-LBP feature extraction

본 논문에서는 CS-LBP 특징 추출을 하기 전 객체의 가려짐에 강인한 모델을 생성하기 위해 그림 2와 같이 파티클을 4x4 로컬 영역으로 분할하고 각 로컬 영역에 대해서 16개의 로컬 CS-LBP 패턴을 추출한다. 이렇게 추출된 각 영역의 패턴들을 연결하여 한 윈도우 당 총 256 크기의 1차원 특징 벡터 $L(x) = \{L_u(x)\}_{u=1\dots m}$ 생성하고 0~1로 정규화 한다. 색상 분포 모델과 마찬가지로 타깃 객체에 대한 질감 분포 모델이 결정되면 추적 대상이 되는 y 좌표 후보 파티클에서도 동일한 질감 모델 $L(y) = \{L_u(y)\}_{u=1\dots m}$ 을 생성하고 정규화 한다.

그림2는 본 논문에서 사용한 로컬 영역 기반의 CS-LBP 특징 벡터를 추출하는 과정을 나타낸다.

III. 파티클 필터와 자기회귀 (autoregressive) 모델을 이용한 객체 추적

색상 분포 모델과 질감 분포 모델이 관측 모델로 생성된 후에, 각 모델은 파티클 필터의 예측 단계에서 각 파티클 들의 관측 우도(likelihood) 를 계산하는데 사용되고, 관측 우도를 기반으로 파티클 들의 가중치를 업데이트 하게 된다. 업데이트 된 가중치에 따라 새로운 타깃이 결정되고 이렇게 결정된 타깃을 중심으로 새로운 파티클들을 재 샘플링 하는 과정을 통해 객체를 반복적으로 추적하게 된다.

3.1. 파티클 필터

파티클 필터^[12]는 SMC (sequential Monte Carlo methods)으로 잘 알려져 있으며, 베이시언 (Bayesian) 확률 분포에 근거하여 이동 물체 추적에 강인한 장점을 가지고 있는 필터이다. 파티클 필터는 유한개의 가중치 파티클 집합

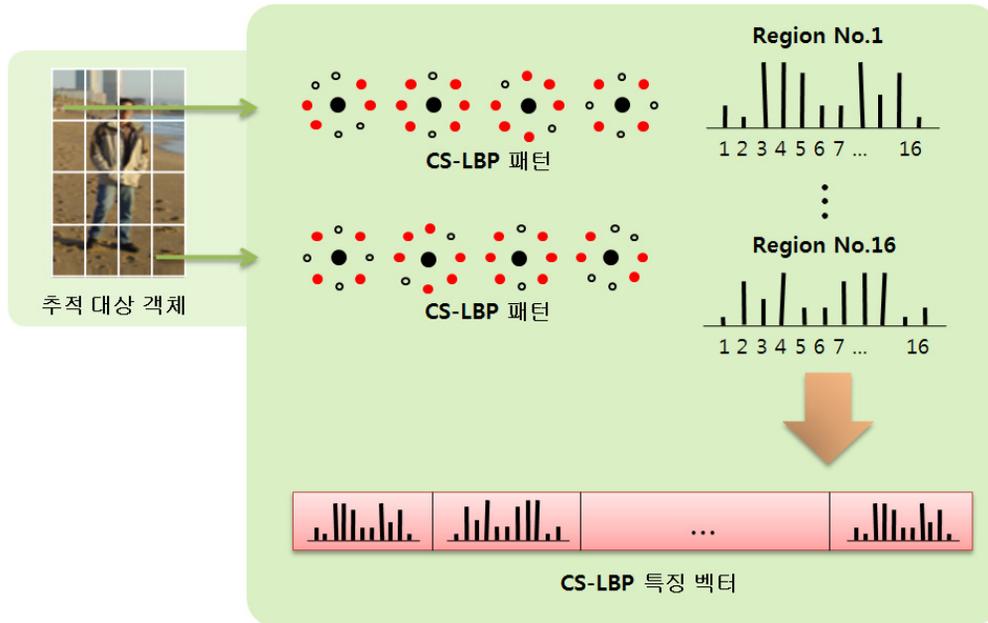


그림 2. 질감 분포 모델 생성을 위한 로컬 CS-LBP 특징 추출 과정

Fig. 2. The Local CS-LBP features extraction process for generating texture distribution model

$S_t = \{s_t^j, j \in \{1 \dots J\}\}$ 으로부터 반복적으로 추적 객체의 사후 확률 분포를 예측한다. 여기서 각 파티클 $s_t^j = (x_t^j, w_t^j)$ 은 상태 (state) 벡터 x_t^j 와 중요가중치 w_t^j 로 구성된다. 파티클 필터는 기본적으로 예측 단계, 업데이트 단계, 재 샘플링 단계로 구성되며 각 프레임마다 이 과정을 반복하며 파티클 집합을 갱신한다.

- 1) 예측 단계 (Prediction): 시간 $t - 1$ 에서 사용 가능한 모든 관찰(observation) $y_{1:t-1} = \{y_1, \dots, y_{t-1}\}$ 이 주어지고, 시간 t 에서 사후 확률을 예측하기 위해 확률 시스템 전이 모델 $p(x_t | x_{t-1})$ 을 식 (4) 에 적용하여 사후 확률을 계산한다.

$$p(x_t | y_{t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1} \quad (4)$$

- 2) 업데이트 단계 (Update): 시간 t 에서, 관찰 y_t 가 사용 가능하면, 상태는 식 (5) 의 베이스 (Bayes) 규칙을 사용하여 업데이트할 수 있다.

$$p(x_t | y_{1:t}) = \frac{p(y_t | x_t) p(x_t | y_{1:t-1})}{p(y_t | y_{1:t-1})} \quad (5)$$

파티클 필터에서 사후 확률 $p(x_t | y_{1:t})$ 은 N 개의 유한개 파티클 $\{x_t^i\}_{i=1, \dots, N}$ 과 각각의 중요가중치 w_t^i 에 의해서 근사된다. 후보 파티클 \tilde{x}_t^i 는 중요 분포 $q(x_t | x_{1:t-1}, y_{1:t})$ 에서 추출되며 파티클들의 가중치는 다음 식 (6) 에 의해서 계산된다.

$$w_t^i = w_{t-1}^i \frac{p(y_t | \tilde{x}_t^i) p(\tilde{x}_t^i | x_{t-1}^i)}{q(\tilde{x}_t^i | x_{1:t-1}, y_{1:t})} \quad (6)$$

- 3) 재 샘플링 단계 (Re-sampling): 재 샘플링 단계에서는 파티클들의 상태가 퇴화하는 것을 막기 위해서 가중치가 높은 파티클들을 중심으로 새로운 파티클들을 다시 추출하는 재 샘플링(re-sampling) 단계를 거친다.

3.2. 관측 모델을 이용한 관측 우도 (likelihood) 추정

관측 모델 (observation model) 은 파티클들의 관측 우도를 측정하는데 사용되는 모델로써 객체 추

적을 위해 가장 중요한 이슈이다. 파티클 필터 추적을 위해 많은 특징 모델들 중에서, 특히 색상 히스토그램은 객체의 부분적 가림이나 비 강체 변환에 일반적으로 좋은 성능을 보여 주었으므로 가장 많이 사용되고 있다^[1]. 하지만 색상 히스토그램은 색상의 지역적 분포가 고려되지 않고 타겟과 유사한 색상을 가진 배경을 잘못 추적 할 수 있다는 단점이 있다. 따라서 색상만을 이용하지 않고 질감 성분을 혼합하여 추적 성능을 향상시키려는 연구가 진행되었다. 질감 성분을 위해서 주로 HOG 특성이 사용되지만 이 방법은 특징 벡터의 차원이 매우 크므로 실시간 검출에 적합하지 않은 단점이 있다.

본 논문에서는 2장에서 설명한 것과 같이 Epanechnikov 커널을 응용하여 파티클의 가운데 영역에 더 많은 가중치를 부여한 HSV 색상 히스토그램과 차원이 작으면서도 HOG, LBP와 유사한 성능을 보여주는 CS-LBP를 파티클의 16개 영역으로부터 추출하고 이 두 개의 모델에 대해 각각 파티클의 관측 우도를 계산함으로써 추적 속도가 빠르고 객체의 가려짐에 강인한 관측 모델을 사용한다.

2.1장과 2.2장에서처럼 두 특징에 대한 관측 모델이 생성된 후에 타겟 특징 모델과 후보 파티클의 특징 모델 간의 거리를 측정한다. 본 논문에서는 유사성 측정에서 KL-distance, Euclidean-distance보다 우수한 것으로 알려진 바타차리야 계수^[13]를 이용한 유사성 측정 방법을 이용한다.

만약 $p^* = \{p_u^*(x_0)\}_{u=1, \dots, m}$ 을 m차원의 타겟 특징 모델이라고 하고, $p(y_t)$ 를 후보 패킷의 특징 모델이라고 한다면, p^* 과 $p(y_t)$ 는 다음 식 (7) 같이 정의될 수 있다.

$$\rho[p^*, p(y_t)] = \left[1 - \sum_{u=1}^m \sqrt{p_u^*(x_0)p_u(y_t)} \right]^{\frac{1}{2}} \quad (7)$$

본 논문에서는 식 (6)을 각 파티클 샘플 x_t^i 에 대한 우도 $p(y_t|x_t^i)$ 로 사용한다.

파티클 필터의 한 형태인 Bootstrap filter^[3]에서는 파티클을 생성과 가중치 계산을 쉽게 하기 위해 식 (6)의 중요분포를 $q(\tilde{x}|x_{1:t-1}, y_{1:t}) = p(x_t|x_{t-1})$ 의 전이모델로

대치하여 사용한다. 이 경우 가중치는 관측 우도 $p(y_t|x_t^i)$ 가 되고 이는 다시 식 (8)과 같이 변형하여 사용할 수 있다.

$$w_t^i = p(y_t|x_t^i) \propto \rho[p^*, p(y_t^i)]^2 \quad (8)$$

본 논문에서는 색상 특징 모델에 대해 파티클간의 관측 우도 $\rho_c[p^*, p(y_t)]$ 를 계산하고, 질감 특징 모델에 대해 파티클 간의 관측 우도 $\rho_t[p^*, p(y_t)]$ 를 계산하도록 하였다. 두 관측 우도의 선형 결합을 위해 식 (8)은 식 (9)와 같이 변형되어 각 파티클들의 가중치로 사용된다.

$$w_t^i \propto (1-\alpha) \cdot \rho_c[p^*, p(y_t^i)]^2 + \alpha \cdot \rho_t[p^*, p(y_t^i)]^2 \quad (9)$$

식 (9)에서 α 는 각 특징 모델을 위한 가중치 파라미터로 본 논문에서는 실험적으로 가장 우수한 성능을 보여준 $\alpha = 0.4$ 로 설정하여 색상 모델에 더 많은 가중치를 부여하였다.

3.3. 파티클 필터링 구현

본 논문에서 사용하는 파티클 필터기반의 추적 알고리즘은 타겟 모델을 초기화 하는 과정과 통계적 추적 시스템을 위한 연속적 몬테카를로 구현 (베이지안 필터링)으로 나뉜다.

3.3.1. 초기 타겟 상태 및 예측

각 프레임별로 타겟의 상태는 식 (10)과 같이 벡터로 표현이 가능 하다.

$$x = [x, y, w, h]^T \quad (10)$$

식 (10)의 파티클 벡터에서 x, y 는 타겟 객체의 위치(중심점)를 나타내고, w, h 는 타겟 객체의 사각형 가로 세로 크기를 나타낸다. 초기 프레임에서는 타겟 사각형의 크기와 위치는 사용자에 의해 수동적으로 결정된다. 본 논문에서는 이후 예측 단계에서 얻어진 가중치를 기반으로 다음 프레임에서 파티클의 위치를 예측하기 위해서 동적 모델인 1차 자기회귀 모델을 사용한다.

식 (11)에서 보는 것과 같이 새로운 파티클의 상태는 이전 상태와 가우시안 랜덤 노이즈가 추가된 상태와의 선형 결합으로 예측된다.

$$\begin{aligned} x_t &= x_{t-1} + G(0, \sigma_x^2) \\ y_t &= y_{t-1} + G(0, \sigma_y^2) \\ w_t &= w_{t-1} + G(0, \sigma_w^2) \\ h_t &= h_{t-1} + G(0, \sigma_h^2) \end{aligned} \quad (11)$$

식 (11)에서 $G(0, \sigma^2)$ 는 각 파티클 들이 랜덤 분포를 갖도록 하기 위한 가우시안 랜덤 노이즈를 나타내며, 가우시안 랜덤 노이즈에 의해 후보 파티클 들의 위치와 크기가 가변적으로 변할 수 있다. 식 (11)에서 사용되는 랜덤 가우시안 분포는 카메라 흔들림과 객체의 크기 변화를 동적으로 반영할 수 있는 중요한 가중치가 된다. 본 논문에서는 파티클의 위치를 위한 가우시안 분포 치를 실험에 의해 $\sigma_x = \sigma_y = 1, \sigma_w = \sigma_h = 0.64$ 로 설정하였다.

3.3.2. 객체 파티클 추적

업데이트 단계에서 식 (9)에 의해 후보 파티클 들의 가중치(w^i)가 결정되면 이 중에서 가중치에 따라 상위 N개의 파티클을 선택하여 t시간에서 타깃의 새로운 상태를 식 (12)와 같이 결정한다. 식 (12)의 파티클 벡터에서 x, y 는 파티클의 중심점을 나타내고, w, h 는 파티클 사각형의 가로 세로 크기를 나타낸다.

$$[x, y, w, h]_t^T = \sum_{i=1}^N w_t^i \cdot [x_t^i, y_t^i, w_t^i, h_t^i]^T \quad (12)$$

이때 현재 새로운 타깃의 상태 중심(x_t, y_t)과 이전 타깃의 상태 중심(x_{t-1}, y_{t-1}) 간의 거리가 임계값 (T) 보다 클 경우 타깃이 다른 물체에 의해 완전히 가려졌거나 잘못된 위치를 검출한 것을 의미하므로 식 (13)과 같이 이전 타깃의 상태 값으로 유지하도록 하였다.

$$[x, y, w, h]_t^T = \begin{cases} [x, y, w, h]_t^T & \text{if } \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} < T \\ [x, y, w, h]_{t-1}^T & \text{otherwise} \end{cases}$$

$$(13)$$

이렇게 추출된 새로운 타깃 상태와 모델을 중심으로 다음번 프레임에서 사용될 파티클을 다시 추출하는 재 샘플링(re-sampling) 단계를 거친다. 재 샘플링 후에 타깃의 새로운 상태 공간은 식 (11)에 의해 결정된다.

그림 3은 파티클 필터링의 전체적인 순서를 나타내고 있다. 그림에서 보는 것과 같이 파티클 필터링은 예측 단계에서 파티클 들의 상태를 예측하고 업데이트 단계에서 파티클들의 관측 우도를 기반으로 각 파티클들의 가중치를 계산하여 새로운 파티클들을 재 샘플링하는 과정을 반복하면서 객체가 사라지거나 사용자에게 의해 객체 추적이 종료될 때까지 객체를 추적하게 된다.

IV. 실험결과

본 논문에서는 제안하는 객체 추적 알고리즘의 성능을 평가하기 위해서 Klein^[6]등의 실험에서 사용한 5개의 동영상을 사용하였다. 각 동영상은 모바일 로봇에 적용하기 위해 움직이는 카메라 환경에서 촬영 되었으며 복잡한 배경 하에서 객체의 가려짐(video1,2,3,4), 색상 및 조명의 변화(video 5)가 연속적으로 발생하는 동영상을 포함하고 있다.

객체 추적 알고리즘의 정량적 성능 평가를 위해 Yin 등^[4]이 제안한 성능 평가 방법을 사용하였다. 그림 4와 같이 사용자가 설정한 객체 영역을 GT (ground-truth) 라고 하고 제안된 알고리즘에 의해 설정된 객체 영역을 ST (system-truth) 라고 한다면, k번째 프레임에서 i객체와 j객체의 공간적 겹침 (A) 은 다음 식 (14)와 같이 정의될 수 있다.

$$A(GT_{ik}, ST_{jk}) = \frac{Area(GT_{ik} \cap ST_{jk})}{Area(GT_{ik} \cup ST_{jk})} \quad (14)$$

실험을 위해 본 논문에서는 (1) RGB히스토그램과 파티클 필터를 이용한 방법, (2) HSV히스토그램과 파티클 필터를 이용한 방법, (3) Epanechnikov 커널 기반의 HSV히스토그램과 파티클 필터를 이용한 방법, (4) 로컬 CS-LBP와 파티클 필터를 이용한 방법, (5) Epanechnikov 커널 기반의 HSV히스토그램과 로컬 CS-LBP의 조합을 파티클 필터에 적용한 제안 방법의 성능을 비교하였다.

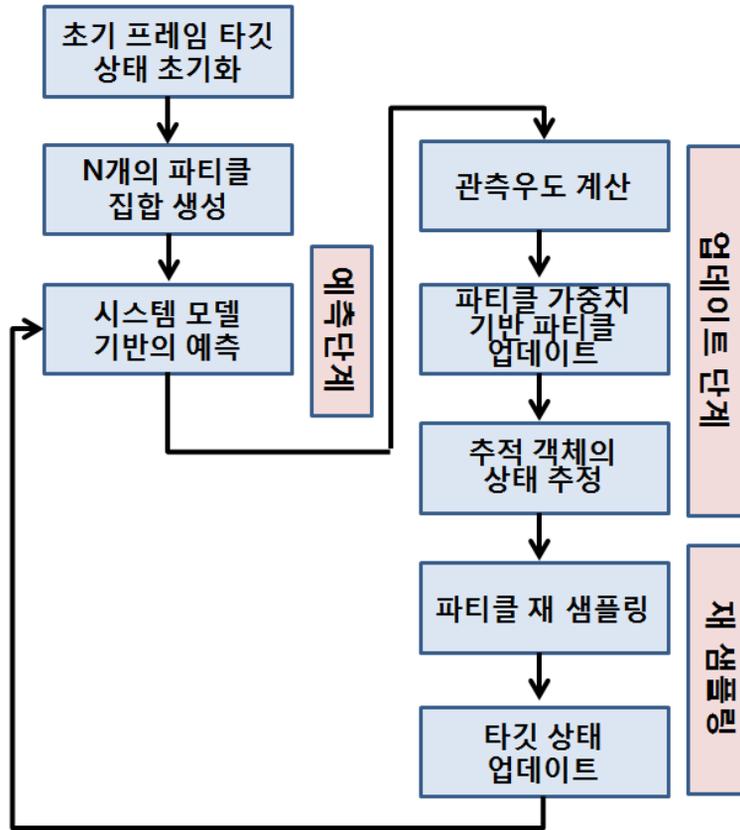


그림 3. 예측단계와 업데이트 단계로 구성된 파티클 필터링 동작 과정
 Fig. 3. The particle filtering algorithm consists of prediction step and update step

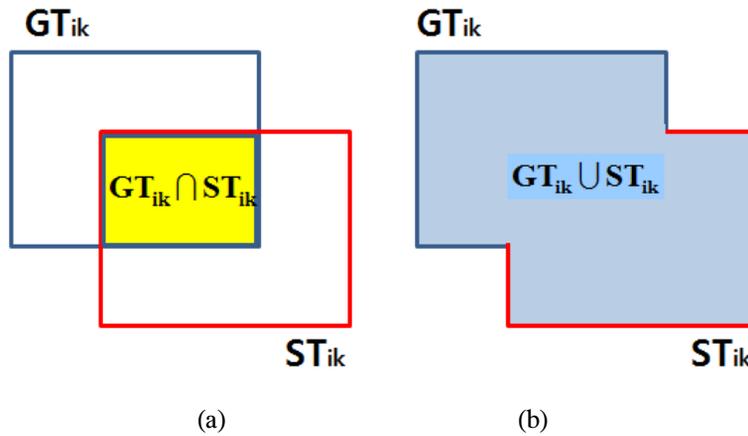
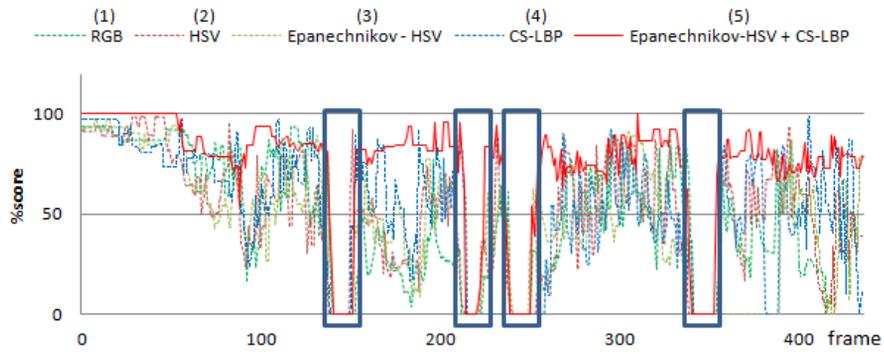


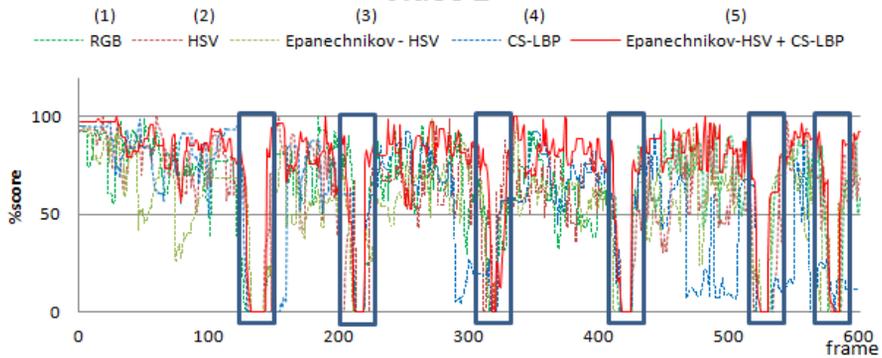
그림 4. 객체간의 공간적 겹침 $A(GT_{ik}, ST_{jk})$ 계산을 위한 (a) $Area(GT_{ik} \cap ST_{jk})$ 영역과 (b) $Area(GT_{ik} \cup ST_{jk})$ 영역
 Fig. 4. $A(GT_{ik}, ST_{jk})$, (a) area of $Area(GT_{ik} \cap ST_{jk})$ and (b) area of $Area(GT_{ik} \cup ST_{jk})$ for calculation of spatial overlapping between objects

Video 1



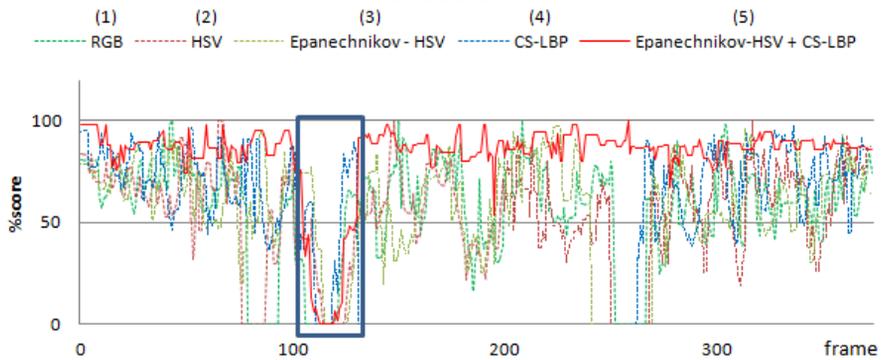
(a) video 1

Video 2



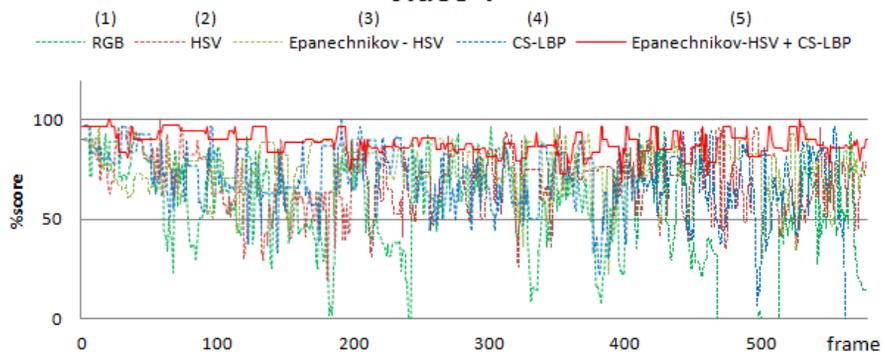
(b) video 2

Video 3

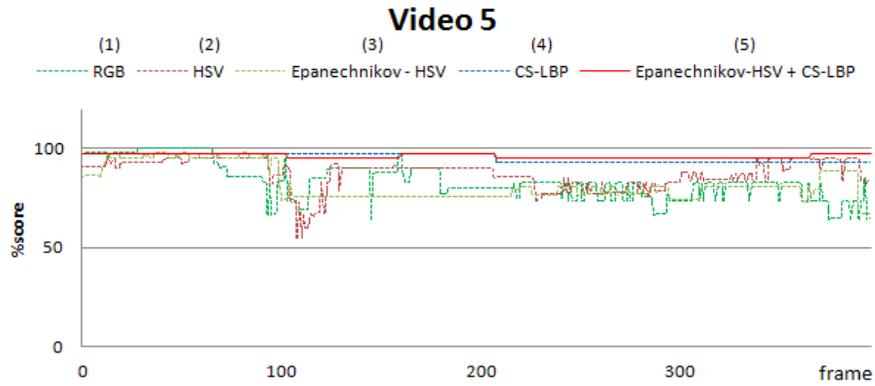


(c) video 3

Video 4



(d) video 4



(e) video 5

그림 5. 각 비디오 별 성능 평가 결과 비교

Fig. 5. Performance evaluation results per video

그림 5는 제안하는 객체 추적 알고리즘과 유사 알고리즘들의 성능 평가 결과를 보여주고 있다. 그림 5에서 그래프의 가로 축은 프레임 번호를 나타내고, 세로축은 픽처클이 포함하고 있는 객체의 검출 스코어를 나타낸다. 그림 5의 (a) ~ (c)에 표시된 사각박스 영역은 객체가 다른 객체에 의해 가려진 부분으로 실제 GT값을 알 수 없으므로 0의 스코어를 갖는 것을 알 수 있다. 성능 평가 결과에서 보듯이, HSV 색상 히스토그램은 RGB에 비해 성능이 우수한 것을 알 수 있다. 이것은 HSV가 RGB에 비해 인간시각과 유사한 색상 값을 가지며 조명 변화에 덜 민감하도록 색상 성분과 밝기 성분을 분리시켰기 때문이다. 마찬가지로 HSV색상 히스토그램을 그대로 사용한 것보다 Epanechnikov 커널 기반의 HSV 히스토그램을 사용하였을 경우 객체의 검출이나 가려짐 부분에서 보다 좋은 성능을 나타낼 수 있다.

로컬 CS-LBP만을 이용한 방법은 역시 좋은 성능을 보여주지만 객체의 형태나 크기가 변할 경우 놓치는 프레임들이 발생한다. 이에 비해 제안된 특징 조합을 사용하였을 경우는 객체의 형태변화는 물론 배경의 복잡함, 객체의 가려짐, 객체의 검출, 조명의 변화에 덜 민감하고 객체의 위치를 정확하게 추적할 수 있었다.

표 2는 각 비디오 영상에서 GT와 ST 사이의 검출 A(GT,ST)에 대한 평균 수치 비교 결과를 보여주고 있다. 그림 5의 객체 검출 영역이 포함된 프레임은 실제 객체가 정확히 추적된 것인지 판단할 수 없으므로 평균 계산에서 제외하였다. 평균 성능에서도 본 논문에서 제안한 방법은 평균 검출(A)율이 86%로 다른 방법들에 비해 우수함을 확인할 수 있다. 하지만 Video2의 경우 비슷한 색상과 크기를 갖는 객체들의 빈번한 가림으로 인해 검출율이 다른 비디오에 비해 상대적으로 떨어졌다.

표 2. 5개의 테스트 비디오에 대한 각 방법별 평균 A(GT,ST) 성능 비교

Table 2. The average A(GT,ST) performance comparison about each method using 5 test videos

비디오 번호	프레임 수	평균 A(GT,ST) x 100 (%)				
		방법 (1)	방법 (2)	방법 (3)	방법 (4)	제안방법
video 1	436	51.8	55.7	51.1	57.7	82.3
video 2	600	66	64.4	59.1	56.8	79.8
video 3	373	57.4	55.1	58.3	44.9	84.9
video 4	578	54.3	69.8	76	68.7	88.4
video 5	396	83.7	87.6	82.2	95.4	96.5
평균		62.64	66.52	65.34	64.7	86.38

그림 6은 본 논문에서 제안하는 알고리즘을 이용하여 객체를 추적한 결과를 보여주고 있다. 그림에서 보듯이 제안하는 알고리즘은 카메라의 떨림이나

배경의 복잡함, 객체의 가려짐, 조명 및 색상의 변화에도 정확한 객체 추적 결과를 보여주고 있음을 알 수 있다.



(a) video 1



(b) video 2



(c) video 3



(d) video 4



(e) video 5

그림 6. 비디오 별 객체 추적 결과
(1번째 열 : 0 프레임, 2번째 열 : 80 프레임, 3번째 열 : 160 프레임, 4번째 열 : 230 프레임)

Fig. 6. The object tracking results in each video

(1st column : 0 frame, 2nd column : 80 frame, 3th column : 160 frame, 4th column : 230 frame)

표 3. 5개의 테스트 비디오에 대한 각 방법별 추적 속도 비교 (fps)
 Table 3. The tracking time comparison in each method using 5 test videos (fps)

비디오 번호	프레임 수	fps				
		방법 (1)	방법 (2)	방법 (3)	방법 (4)	제안방법
video 1	436	10.4	9.9	9.9	9.9	10.1
video 2	600	8.7	8.4	8.6	8.7	8.4
video 3	373	10.6	9.8	10.1	9.8	9.6
video 4	578	11.6	11.8	11.7	11.5	11.6
video 5	396	10.1	9.7	9.7	9.6	9.4
평균		10.3	9.9	10	9.9	9.8

본 논문에서 제안한 알고리즘의 추적 속도를 측정하기 위해 5개 비디오에 대해서 초당 프레임 (fps:frame per second)을 측정하였다. 비교 알고리즘들은 Windows 7 운영체제의 Visual C++ 2008 버전에서 개발 되었으며, 동일한 Intel I-7 2.67GHz CPU환경에서 테스트 되었다.

표3 에서 보는 바와 같이 본 논문에서 제안한 방법은 평균 9.8프레임의 검출 속도를 가짐을 알 수 있다. 이는 평균 10.3프레임의 검출 속도를 갖는 기존의 방법 (1)에 비해 초당 0.5프레임 느리지만 이는 실시간 객체 추적에는 충분한 속도이다. 특히, 검출 성능에서는 RGB색상 히스토그램보다 약 20%의 향상된 검출 율을 보여주었으므로 모바일 로봇과 같은 움직임 카메라 환경에서는 기존의 방법 보다 적합한 알고리즘이라고 할 수 있다.

V. 결 론

본 논문에서는 파티클 필터와 HSV 색상 모델, 로컬 CS-LBP 질감 모델을 조합하여 실시간으로 객체를 추적하는 알고리즘을 제안하였다. 먼저, N개의 파티클을 생성하고, 각 파티클 별로 HSV와 로컬 CS-LBP특징을 추출 하였다. 그리고 타깃의 특징과 N개 파티클 특징 간의 관측 우도를 각 특징 모델별로 바타차리야 거리를 이용하여 측정 하였다. 이렇게 측정된 관측 우도의 선형 결합으로 가중치를 갱신하고 가중치가 가장 높은 파티클을 새로운 타깃으로 설정 한 뒤 다음 프레임에서 사용할 파티클들을 재 샘플링 하였다.

본 논문에서는 파티클 필터를 사용하여 처리 속도를 향상 시켰으며, 색상 정보와 질감 정보의 사용으로 보다 정확하고 강인한 객체 추적 성능을 보여주

었다.

앞으로의 연구에서는 관측 우도를 계산함에 있어서 바타차리야 거리를 이용하는 것이 아니라 패턴 분류기를 실시간 학습하여 유사도를 비교함으로써 환경의 변화나 객체의 크기 및 자세 변화에도 변함 없는 성능을 보여줄 수 있는 알고리즘을 개발하고자 한다.

감사의 글

본 논문은 연구재단의 ‘대구과학연구단지육성사업 융합기술 기초R&D 지원사업(2011)’의 연구기금으로 수행 되었음.

참 고 문 헌

- [1] C. Yang, R. Duraiswami, and L. Davis "Fast Multiple Object Tracking via a Hierarchical Particle Filter," *International Conference on Computer Vision*, Vol. 1, pp.212-219, 2005.11
- [2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vo. 25, No. 5, pp. 564-577, 2003.
- [3] A. Doucet, N. D. Freitas, and N. Gordon, "Sequential Monte Carlo Methods in Practice," *Springer-Verlag*, New York, 2001
- [4] K. Deguchi, O. Kawanaka, and T. Okatani, "Object tracking by the mean-shift of regional color distribution combined with the particle-filter algorithm," *17th International Conference on Pattern Recognition*, Vol. 3, pp.

506-509, 2004.

[5] M. Z. Islam, C. M. Oh, and C. W. Lee, "Video Based Moving Object Tracking by Particle Filter", *International Journal of Signal Processing, Image Processing and Pattern*, Vol. 2, No. 1, pp.120-132, 2009.

[6] D. A. Klein, D. Schulz, S. Frintrop, and A. B. Cremers, "Adaptive Real-Time Video-Tracking for Arbitrary Objects", *IEEE/RSJ International Conference of Intelligent Robots and Systems*, pp. 772-777, 2010.

[7] D. Sidibe, D. Fofi, and F. Meriaudeau, "Using visual saliency for object tracking with particle filters," *18th European Signal Processing Conference*, pp. 1-5, 2010.

[8] Z. H. Khan, I. Y. Gu, and A. G. Backhouse, "Robust Visual Object Tracking Using Multi-Mode Anisotropic Mean Shift and Particle Filters," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 21, pp.74-87, 2011.

[9] D. P. Chau, F. Bremond, M. Thonnat, and E. Corvee, "Robust Mobile Object Tracking Based on Multiple Feature Similarity and Trajectory Filtering," *International Conference on Computer Vision Theory and Applications*, pp. 1-7, 2011.

[10] K. Nummiaro, E.K. Meier, and L.V. Cool, "An adaptive color-based particle filter," *Image and Vision Computing*, Vol. 21, No. 1, pp. 99-110, 2003.

[11] B. C. Ko, S. H. Kim, J. Y. Nam, "X-ray Image Classification using Random Forests with Local Wavelet-Based CS-Local Binary Patterns," *Journal of Digital Imaging*, Vol. 24, pp. 1141-1151, Nov. 2011

[12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, Vol. 5, No. 2, pp. 174-188, 2002.

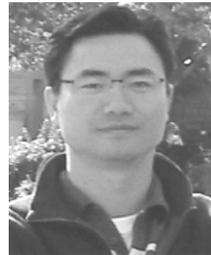
[13] D. Comaniciu, V. Ramesh, and P. Meer. "Real-Time Tracking of Non-Rigid Objects using Mean Shift," *IEEE International Conference on Computer Vision and Pattern*

Recognition, pp. 142 - 149, 2000.

[14] F. Yin, D. Markris, and S. Velastin, "Performance evaluation of object tracking algorithms," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 1-8, 2007.

고 병 철 (Byoung-Chul Ko)

정회원



1998년 경기대학교 전자계산학과 졸업(이학사)

1998년~2000년 연세대학교 대학원 컴퓨터과학(공학석사)

2000년~2004년 연세대학교 대학원 컴퓨터과학(공학박사)

2004년 3월~2005년 8월 삼성전자 통신연구소 책임 연구원

2005년 9월~현재 계명대학교 컴퓨터공학과 부교수
<관심분야> 내용기반 영상검색, 컴퓨터 비전 및 패턴인식

남 재 열 (Jae-Yeal Nam)

정회원



1983년 경북대학교 전자공학과 졸업(공학사)

1985년 경북대학교 대학원 전자공학(공학석사)

1991년 University of Texas at Arlington 전기공학(공학박사)

1985년 5월~1987년 7월 한국전자통신연구소 연구원

1991년 9월~1995년 2월 한국전자통신연구소 선임 연구원

1995년 3월~현재 계명대학교 컴퓨터공학과 교수
<관심분야> 영상압축, 영상통신, 멀티미디어 시스템

곽 준 영 (Joon-Young Kwak)

준회원



2011년 계명대학교 컴퓨터공학과 졸업(공학사)

2011년 계명대학교 대학원 입학

<관심분야> 산물 연기 감지, 휴먼 객체 추적