

# 물체 특징과 실시간 학습 기반의 파티클 필터를 이용한 이동 로봇에서의 강인한 물체 추적

## Robust Object Tracking in Mobile Robots using Object Features and On-line Learning based Particle Filter

이형호, 최학남, 김형래, 마승완, 이재홍, 김학일\*  
(Hyung-Ho Lee<sup>1</sup>, Xuenan Cui<sup>2</sup>, Hyoung-Rae Kim<sup>2</sup>, Seong-Wan Ma<sup>2</sup>, Jae-Hong Lee<sup>2</sup>, and Hak-Il Kim<sup>2</sup>)

<sup>1</sup>LIG Nex1

<sup>2</sup>Inha University

**Abstract:** This paper proposes a robust object tracking algorithm using object features and on-line learning based particle filter for mobile robots. Mobile robots with a side-view camera have problems as camera jitter, illumination change, object shape variation and occlusion in variety environments. In order to overcome these problems, color histogram and HOG descriptor are fused for efficient representation of an object. Particle filter is used for robust object tracking with on-line learning method IPCA in non-linear environment. The validity of the proposed algorithm is revealed via experiments with DBs acquired in variety environment. The experiments show that the accuracy performance of particle filter using combined color and shape information associated with on-line learning (92.4 %) is more robust than that of particle filter using only color information (71.1 %) or particle filter using shape and color information without on-line learning (90.3 %).

**Keywords:** object tracking, particle filter, HOG, color histogram, IPCA, mobile robots

### 1. 서론

최근 로봇은 다양한 분야에서 필요와 요구에 따라 그 개발 분야도 다양화 되어가고 있다. 자동화 공장에서의 로봇부터 지능화된 이동 로봇까지 그 범위는 점차 늘어나고 그와 더불어 카메라를 이용한 컴퓨터 비전 기술의 요구도 늘어나고 있는 추세이다. 로봇에서의 컴퓨터 비전 기술은 로봇의 시각의 역할을 수행한다. 특히 카메라나 다른 비전 센서로부터 획득된 데이터에서 특정 물체를 추적하는 것은 그 정보를 제어부에 전달함으로써 다른 부가적인 기능을 수행하는데 큰 역할을 하며 이를 위해 높은 정확성을 가진 추적이 요구되었다. 또한 응용의 목적에 따라 추적해야 하는 대상과 대상이 가지는 특징, 주변 환경의 요소들이 다르기 때문에 적절하게 추적이 수행될 수 있도록 고려해야 하는 사항도 많다.

자율 이동 로봇에서의 추적은 다양한 환경에서 이루어져야 하기 때문에 많은 문제가 존재하게 된다. 일반적으로 카메라의 떨림, 조명의 변화, 물체의 형태의 변화, 복잡한 배경, 다른 물체에 의한 가려짐과 같은 많은 외부적 요인에 의해서 성능에 많은 영향을 미치게 된다. 이러한 문제점을 해결하기 위해 다양하게 변화하는 외부 환경에 대해서 강인하게 물체를 추적할 수 있도록 많은 연구가 진행되고 있다[1].

물체를 추적하기 위해서는 영상 내의 물체의 특징을 이용하여야 한다. 이동 로봇과 같은 환경의 동적 카메라에서 물체의 특징을 이용하는 방법에는 크게 두 가지가 있다. 첫 번째는 학습을 통해 형성된 정보를 이용하여 영상 내에서 직접적으로 물체를 검출하여 추적하는 방법이다[2-4]. 이는 다양한 환경에서 많은 학습을 수행하여야 좋은 성능을 기대할 수 있으며, 시간적으로 많은 노력이 필요한 방법이다. 두 번째는 물체 영역에 대해서 확률적으로 물체를 표현하여 추적하는 방법이다[5-10]. 초기화를 통해 추적할 물체의 정보를 특징화하고 이를 통해 연속적인 프레임에서 비슷한 특징을 가지는 영역을 통해 물체를 추적하는 방법이다. 최근 두 방법을 혼합하여 다양한 응용에서 물체를 추적하는 연구가 진행되고 있으며, 이는 기존의 방법보다 외부적인 요인에 강인한 성능을 보이고 있다[4,11-20].

환경에 맞는 물체 추적을 위해서는 물체의 특징을 정의하는 것도 중요한 요소이다. 색상 히스토그램은 기존의 많은 연구에서 물체 추적을 위해 사용되어왔던 특징이다[4-9,14]. 하지만 조명의 변화가 빈번한 환경에서는 물체를 표현하기에 충분하지 못하다. 최근 물체를 효과적으로 표현하고자 HOG (Histogram of Oriented Gaussian) 방법이 연구되었다[4,14,16,17]. 영상 내에서 물체의 형태를 표현하기에 적합하며, 조명의 변화에 강인한 성질을 나타내어 많은 물체 추적 응용에서 사용되고 있다.

본 논문에서는 측면 카메라를 이용한 이동 로봇에서 발생하는 빠르게 변화하는 배경, 조명의 변화, 물체 형태의 변화 등의 문제점에 강인한 물체 추적 알고리즘을 제안한다. 물체의 특징을 보다 세밀하게 표현하기 위해 색상 히스토그램과 HOG 서술자를 혼합하여 이용하였으며, 비선형적인 환경에

\* 책임저자(Corresponding Author)

논문접수: 2012. 1. 1., 수정: 2012. 3. 27., 채택확정: 2012. 5. 3.

이형호: LIG Nex1(hyung.ho.lee@lignex1.com)

최학남, 이재홍, 김학일: 인하대학교 정보공학과

(xncui@vision.inha.ac.kr/jaehong@vision.inha.ac.kr/hikim@inha.ac.kr)

김형래, 마승완: 인하대학교 로봇공학과

(hrkim@vision.inha.ac.kr/swma@vision.inha.ac.kr)

※ 본 연구는 2010 지식경제부 산업원천기술개발사업(1148122210)의 지원에 의하여 연구되었음.

적합한 추적 알고리즘인 파티클 필터(Particle filter)를 통해 물체를 추적하였다. 또한 추적된 물체의 정보를 실시간으로 학습하고 지속적으로 변화하는 물체의 주요 성분을 이용하여 기존의 방법보다 추적 성능을 높였다.

본 논문의 구성은 다음과 같다. 제 I 장에서는 본 논문에 대한 서론이며, 제 II 장에서는 물체 추적 알고리즘의 전반적 구성을 소개한다. 제 III 장에서는 물체 추적을 위한 특징 추출에 대해 소개하고, 제 IV 장에서는 IPCA (Incremental Principal Component Analysis)를 이용한 실시간 학습 방법을 소개한다. 제 V 장에서는 물체 특징을 혼합하고 실시간 학습을 이용한 파티클 필터 기반의 알고리즘을 제안한다. 제 VI 장에서는 실험을 통해 제안된 알고리즘의 성능을 검증하며, 제 VII 장에서는 연구에 대한 결론 및 향후 과제에 대해 기술한다.

**II. 물체 추적 알고리즘의 전반적 구성**

측면 카메라를 이용한 이동 로봇에서의 물체 추적은 이동 로봇의 움직임과 물체의 움직임이 독립적이며, 추적하는 물체는 그 형태를 지속적으로 유지하지 않는다. 또한 실제 환경에서 물체는 빛의 변화에 따라 물체가 가지는 색상의 변화를 가져올 수 있다. 따라서 이러한 영향에 보다 강인하게 하기 위해서는 선형적이지 않은 움직임을 추적할 수 있어야 하며, 물체의 형태와 색상의 변화에 따라 지속적으로 물체를 모델링 하여야 한다[14].

이동 로봇의 이동과 독립적인 물체의 이동으로 발생하는 비선형적인 움직임을 보다 효과적으로 추적하기 위해 파티클 필터를 이용하여 물체를 추적한다. 또한 물체의 형태와 색상 특징을 추출하여 물체를 모델링하며, 연속되는 프레임 영상에서의 물체의 형태와 색상을 강인하게 측정하기 위해 실시간 학습 방법을 사용한다[15-19]. 실시간 학습 방법을 사용하면 형태와 색상이 변화하는 물체에서의 대표적인 성분을 통해 모델링 할 수 있다는 점에서 물체 추적에 보다 효과적이다. 제안된 알고리즘의 순서도는 그림 1과 같다.

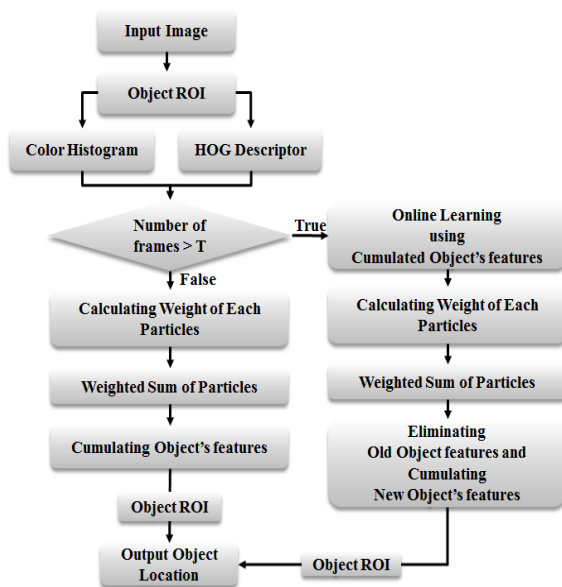


그림 1. 이동 로봇에서의 물체 추적 순서도.  
Fig. 1. Object tracking in mobile robots flow chart.

입력된 영상에서 추적하고자 하는 물체의 ROI를 선택하고, ROI의 영역을 16x24의 영상으로 크기를 재설정 한 후, ROI 내에서 형태 특징을 나타낼 수 있는 HOG 서술자와 색상 특징을 나타내는 색상 히스토그램을 형성한다. 영상의 크기를 재설정하는 이유는 물체의 특징의 차원을 줄이고, 주요 특징 성분만 계산하기 위함이다. 형성된 두 특징은 실시간 학습이 진행되기 전까지의 T 프레임까지 물체를 표현하는 초기 특징으로 사용된다. 초기화를 통해 영상 전체에 분포된 파티클 영역에서 HOG 서술자와 색상 히스토그램을 각각 계산하고, 이를 이용하여 초기 특징과의 유사성을 통해 각 파티클의 가중치를 부여한다. 각 파티클의 가중치에 따라 가중 합을 구하여 현재 프레임에서의 물체의 위치를 계산하며, 물체의 위치에서의 색상과 형태 특징은 실시간 학습을 위해 초기 정해 놓은 T 프레임까지 누적을 한다.

T 프레임 이후부터는 누적된 형태와 색상 특징을 실시간 학습 방법 중 하나인 IPCA를 이용하여 대표적인 성분을 계산한다[15,16]. 이 성분은 앞으로 각 파티클에서 계산된 색상과 형태 특징을 재구성하는데 사용된다. 현재 프레임에서 각 파티클에서 계산된 색상과 형태 특징과 재구성된 색상과 형태 특징과의 유사성을 계산하여 각 파티클에 가중치를 부여한다. 이후 T 프레임 이전의 계산과 같이 파티클들의 가중 합을 통해 물체의 위치를 계산하며, 이 위치에서의 색상과 형태 특징을 누적함과 동시에 오래된 특징을 제거한다. 제안된 알고리즘의 T는 40으로 정의하였다.

**III. 물체 추적을 위한 특징 추출**

이동 로봇에서의 물체 추적은 추적하고자 하는 물체의 특징을 환경에 맞게 정의해야 하고 물체를 잘 표현해야 하는 두 가지를 만족해야 한다. 환경에 맞게 물체의 특징을 정의하기 위해서는 그 환경이 먼저 고려되어야 한다. 환경이 복잡하지 않고 추적하고자 하는 물체가 구분하기 명확하다면 색상이나 형태 혹은 다른 종류의 단일 특징만으로 물체를 기술하는 것이 적절하다. 하지만 이동 로봇에서의 환경과 같이 다양하고 동적인 배경을 가지는 환경이라면 물체가 가지는 특징을 정의하기란 쉽지 않다. 따라서 다중의 특징을 이용하는 것이 적합하다. 물체가 가지는 특징을 적절하게 정의하기 위해서 본 논문에서는 색상 히스토그램과 HOG를 이용한다.

**1. Color histogram**

컴퓨터 비전에서 가장 많이 사용되는 물체를 표현하는 특징은 색상 히스토그램이다[5-9]. 색상 히스토그램은 RGB, HSV, YCbCr 등 다양한 채널 영역에서 계산해 낼 수 있으며, 이를 기반으로 많은 응용 분야에서 사용되고 있다. 색상 히스토그램을 계산해 내기 위해서는 몇 개의 채널을 사용할 것인지와 몇 개의 양자화된 구간으로 나눌 것인지가 중요하다. 채널이 많이 포함되어 있으면 원하는 물체 혹은 영상을 표현하는데 많은 정보를 가지는 대신 실제 연산 과정에서 많은 처리 시간을 소요하게 한다. 그리고 양자화 시킨 구간인 저장소(Bin)의 개수에 따라 중요 색상만을 알아낼 수도 있고 혹은 물체를 표현하는 모든 색상을 표현할 수 있다. 일반적으로 관심 영역으로 검출되는 영역은 다소 배경의 영역을 포함되어 있으므로 이와 관련하여 배경 영역의 영향을 줄이고자 커널(Kernel)을 이용하여 색상 히스토그램을 계산하였다

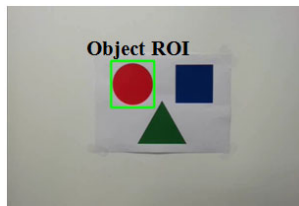
[7]. 이는 식 (1)과 같이 표현된다.

$$q_u = \sum_{i=1}^n k(\|x_i^*\|^2) \delta[c(x_i) - u] \quad (1)$$

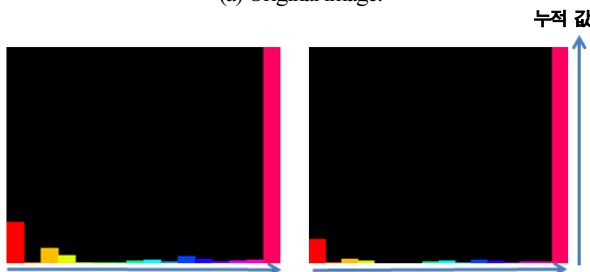
$$k(\|x_i^*\|^2) = 1 - r^2$$

$q_u$  는 히스토그램에서 나타내는 각 저장소에 누적된 색상의 개수를 의미하며  $u$  의 범위는 한 개의 채널을 몇 개의 저장소로 나누었는지를 정의한다.  $\delta[c(x_i) - u]$  은 영상에서의 픽셀의 위치  $x_i$  에서의 색상 값이 해당 저장소의 범위에 속하였을 때 개수를 누적함을 나타낸다.  $k(\|x_i^*\|^2)$  는 물체를 표현한 ROI내 중심 픽셀에서 가장 먼 픽셀까지의 거리를 1로 정규화한 값에서 계산하고자 하는 픽셀  $x_i^*$  와 중심 픽셀간의 정규화된 거리를 뺀 값을 나타낸다. 그림 2는 HSV 모델 색상(Hue) 채널에서 식 (1)을 이용하여 물체의 관심영역에서의 색상 히스토그램과 커널을 이용한 가중된 색상 히스토그램의 결과를 나타낸다.

물체의 색상 정보를 보다 정밀하게 나타내기 위해서는 1차원 색상 히스토그램보다 3차원 색상 히스토그램을 이용하여야 한다. 따라서 HSV모델에서 색상(hue), 채도(saturation), 밝기(value)를 모두 이용하도록 구성하였으며, 각 채널의 차



(a) Original image.



(b) Color histogram.

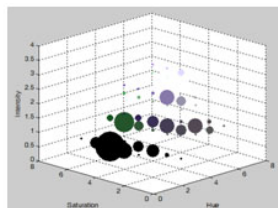
(c) Weighted color histogram.

그림 2. 물체 관심영역에서의 색상 히스토그램과 가중된 색상 히스토그램 비교.

Fig. 2. (a) Comparison between color histogram and weighted color histogram in object ROI.



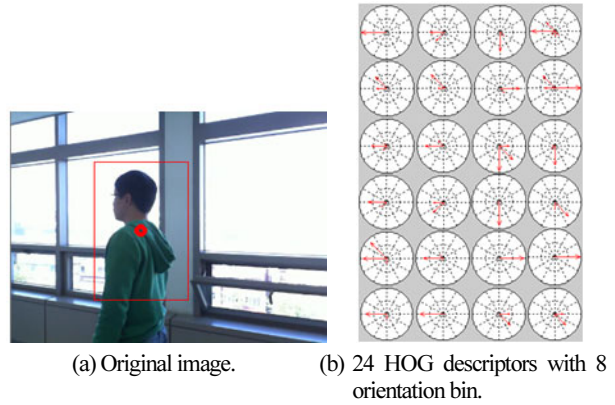
(a) Original image.



(b) 3D color histogram in HSV (8x8x4).

그림 3. HSV모델에서의 3D 색상 히스토그램 (8x8x4).

Fig. 3. 3D color histogram in HSV (8x8x4) in HSV model.



(a) Original image.

(b) 24 HOG descriptors with 8 orientation bin.

그림 4. 8개의 방향성을 가지는 24(6x4)개의 HOG 서술자.

Fig. 4. 24 HOG descriptor with 8 orientation bin.

원은 8x8x4를 이용하였다. 이동 로봇의 환경에서는 조명의 변화가 빈번히 발생할 수 있으므로 차원을 적게 하여 조명의 변화에 민감하지 않도록 구성하였다. 그림 3은 이동 로봇의 측면 카메라를 통해 입력 받은 영상에서 식 (1)을 이용하여 물체를 표현하는 관심영역의 3D 색상 히스토그램을 표현한 결과이다.

## 2. HOG (Histogram of Oriented Gradient)

최근 영상에서 사람을 검출 및 추적하기 위해 많이 쓰이는 대표적인 특징은 방향성 기울기 히스토그램 HOG이다[4, 14, 16, 17]. 기울기 히스토그램은 이름에서와 같이 영상 내에서 기울기(gradient)와 각각의 기울기가 가지는 방향성(orientation)을 이용하여 서술자로 표현하는 방식이다. 기울기히스토그램을 계산하기 위해 RGB 3채널의 영상 자체를 이용하는 방법도 있지만, 차원의 효율성을 위해 1채널의 그레이(gray) 영상을 이용하는 방법을 이용하였다. 그림 4는 실제 영상에서의 ROI내의 물체를 표현하는 HOG 서술자를 나타낸다. 그림 4의 왼쪽 그림에서의 물체 ROI를 6x4로 중복되지 않는 격자 영역으로 분할하고, 각 격자에서의 크기와 방향성에 따라 값을 누적하여 각각의 서술자 표현하고, 6x4의 서술자들을 모두 정규화하여 오른쪽 결과와 같이 표현하였다. 그림 4의 결과와 같이 HOG 서술자를 계산하는 방법은 다음과 같다.

1. 윈도우 크기 5x5의 가우시안 필터(Gaussian filter)를 이용한 잡음이 제거된 영상 획득
2. 획득된 영상의 픽셀  $I(x, y)$  에서 X와 Y방향으로의 각각 1D 마스크를 이용한 기울기(Gradient)  $g_x(x, y)$  와  $g_y(x, y)$  계산

$$g_x(x, y) = I(x+1, y) - I(x-1, y) \quad (2)$$

$$g_y(x, y) = I(x, y+1) - I(x, y-1)$$

3. 영상내의 각 픽셀 당 기울기의 크기(magnitude)인  $m(x, y)$  및 방향성(Angle)  $\theta(x, y)$  을 계산

$$m(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2} \quad (3)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{g_y(x, y)}{g_x(x, y)} \right)$$

4. 방향성 성분을 0~360도로 만들기 위해 방향성이 0보다 작을 경우  $2\pi$  를 더하여 계산

$$\tilde{\theta}(x, y) = \begin{cases} \theta(x, y) + 2\pi & \text{if } \theta(x, y) < 0 \\ \theta(x, y) & \text{otherwise} \end{cases} \quad (4)$$

5. 4x4 격자(grid) 안에서 8개의 방향성 저장소  $u$  로 누적과 정규화(normalization)을 이용한 서술자(descriptor) 형성

$$D_u = \frac{1}{f} \sum_{i=1}^n m_i(x, y) \delta(\tilde{\theta}_i(x, y) - u), \quad u = 1, \dots, 8 \quad (5)$$

**IV. IPCA를 이용한 실시간 학습**

이동 로봇에서 측면 카메라를 통해 추적하고자 하는 물체는 매 프레임마다 다른 형태를 취하고 있으며, 또한 실제 환경 구간마다 다른 빛의 양 때문에 조명의 영향으로 다른 색상을 나타나게 된다. 따라서 이를 보다 효과적으로 해결하기 위해서는 실시간으로 물체의 정보를 갱신하여 물체를 추적할 때 사용하도록 해야 한다. 이러한 문제를 해결하기 위해 다양한 실시간 학습 방법이 연구되고 있으며, 본 연구에서는 패턴인식과 관련하여 많은 응용에서 사용되고 있는 PCA 기반 학습 방법을 사용하였다. 본래 PCA는 실시간이 아닌 사후시간 학습으로 많이 사용되는 방법인데, 위와 같은 문제들을 해결하기 위한 요구에 따라 실시간을 위한 IPCA (Incremental PCA) 방법이 연구되었다[15,16].

IPCA는 PCA와 같이 특정 값들의 주성분을 분석하는 것이 주요 목표이다. 그림 5는 주성분 분석을 위한  $U$  와  $\Sigma$  를 구하는 과정을 나타낸다. 주성분 분석을 위해서는 특정 시간에 누적된 특징 값들을 정의하여 구성해야 한다. IPCA에서 사용되는  $d \times n$  의 행렬  $A = \{I_1, \dots, I_n\}$  는 매 프레임 영상에서 계산된 물체의 특징인 각 열의  $d$  차원 벡터  $I_i$  로 구성된다. 제안된 알고리즘에서는 색상 히스토그램과 HOG 서술자 모두  $n = 40$  으로 정의하였고, 각 특징의 차원을  $d$  로 정의하였다.  $A$  는 SVD (Singular Value Decomposition)을 이용하여  $A = U\Sigma V^T$  로 표현될 수 있다.  $B$  는  $A$  이후의 프레임에서 관측된 물체의 특징들로 만든  $d \times m$  행렬이다. 제안된 알고리즘에서의  $m = 1$  로 정의하였다. 그리고 두 행렬  $A$  와  $B$  에

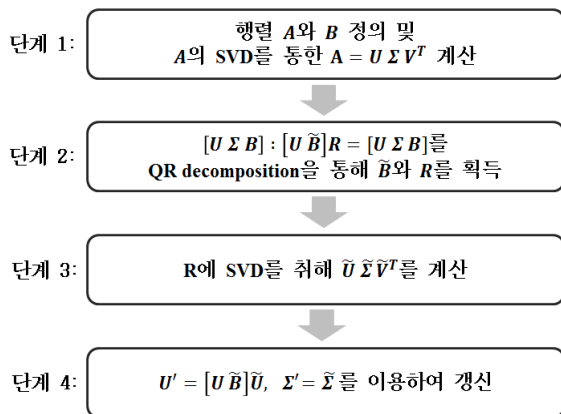


그림 5. 실시간 학습을 위한 IPCA 과정.  
Fig. 5. Process of IPCA for on-line learning.

서의 중요 성분을 분석하기 위해서  $[A \ B] = U\Sigma V^T$  로 나타낼 수 있으며, 식 (6)은  $[A \ B]$  의 주요 성분을 분석하기 위해 세분화된 행렬의 곱으로 표현한 식이다.

$$[A \ B] = [U \ \tilde{B}] \begin{bmatrix} \Sigma & U^T B \\ 0 & B^T B \end{bmatrix} \begin{bmatrix} V^T & 0 \\ 0 & I \end{bmatrix} \quad (6)$$

여기서  $\tilde{B}$  는  $U$  에 직교인  $B$  의 요소를 나타내며, 식 (6)의 오른쪽 행렬들 중 가운데 행렬을  $R$  로 표현한다.  $R$  은 SVD를 통해  $R = \tilde{U} \tilde{\Sigma} \tilde{V}^T$  로 다르게 표현할 수 있다. 식 (7)은 식 (6)을  $R$  의 SVD를 이용한 표현에 따라 다시 한번 재 구성한 것을 나타낸다.

$$[A \ B] = U\Sigma V^T = ([U \ \tilde{B}] \tilde{U}) \tilde{\Sigma} \left( \tilde{V}^T \begin{bmatrix} V^T & 0 \\ 0 & I \end{bmatrix} \right) \quad (7)$$

식 (7)을 이용하여 최종적으로 구하고자 하는  $U$  와  $\Sigma$  를 계산할 수 있다.  $U$  와  $\Sigma$  는  $A$  에서의  $U$  와  $\Sigma$  가  $B$  의 정보가 더해져 갱신된 값을 나타낸다. 따라서 이를 이용하면 연속되는 프레임 안에서 물체의 특징을 효과적으로 갱신할 수 있다.

**V. 파티클 필터를 이용한 물체 추적**

파티클 필터는 최근 들어 많은 연구가 진행되고 있는 알고리즘이며, 다양한 분야의 시스템에서 사용되고 있다[4,10,11, 14,19]. 시간에 따라 변화하는 시스템은 항상 잡음이나 모호한 값들이 존재하게 되는데, 이러한 값들을 제거하고 확률적으로 가능성이 높은 값을 추정하여 시스템의 상태를 파악하는 방법으로써 그 효과가 입증되어 있다. 가장 많이 사용하는 분야는 로봇에서의 위치 추정(robot localization), 영상에서의 물체 추적(object tracking), 사람 포즈 인식(pose recognition), 게임에서의 인공 지능(artificial intelligence) 등이 있다. 커다란 문맥에서 파티클 필터는 bayesian framework 안에서 실시간 학습을 위해 monte carlo sampling 방법을 이용한 recursive bayesian filter를 구현하는 기술이다. 필터를 구현하기 위해 많은 양의 파티클들을 사용하게 되는데 각각의 파티클들은 상태 변수(state variable)와 가중치(weight)를 포함하고 있다. 이러한 파티클들의 집합을 이용하여 영상에서의 현재 상태의 물체를 표현하는 방법이 파티클 필터이다. 파티클 필터는 다양한 종류가 있지만, 본 논문에서는 SIR (Sequential Importance Resampling) 방법의 condensation 알고리즘을 사용하였다[11].

파티클 필터를 사용하기 위해서는 먼저 상태 변수를 정의하여야 한다. 대부분의 영상을 이용한 물체 추적에서는 상태 변수를 식 (8)과 같이 정의하여 사용한다.

$$X = [x, y, w, h, \dot{x}, \dot{y}]^T \quad (8)$$

$x$  와  $y$  는 영상에서의 물체의 위치를 표현하기 위한 변수이며 각 파티클이 있는 위치를 나타낸다.  $w$  와  $h$  는 영상에서의 물체의 폭과 크기의 정보를 표현하기 위한 변수이며 각 파티클의 폭과 크기를 나타낸다. 또한  $\dot{x}$  와  $\dot{y}$  는 물체가 이동하는 속도를 나타내는 변수이며 각 파티클이 가지고 있는

속도 값을 나타낸다. 파티클 필터에서는 이러한 상태 변수를 이용하여 과거의 상태 변수를 이용하여 현재의 상태를 예측하며,  $t$  시간의 상태 변수를 갱신하기 위해 이전 시간  $t-1$ 에서의 상태 변수에 측정 행렬 곱하고 잡음을 더함으로써 시스템 모델(system model)을 식 (9)와 같이 정의한다.

$$\begin{bmatrix} x_t \\ y_t \\ w_t \\ h_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ w_{t-1} \\ h_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \gamma_{width} \\ \gamma_{height} \\ \gamma_{\dot{x}} \\ \gamma_{\dot{y}} \end{bmatrix} \quad (9)$$

잡음 행렬의  $\gamma_{width}$  와  $\gamma_{height}$  는 물체의 폭과 크기에 따라 형성된 잡음의 값을 의미하며,  $\gamma_{\dot{x}}$  와  $\gamma_{\dot{y}}$  는 물체의 속도에 따라 포함되는 잡음 값이다. 잡음을 발생시키기 위해  $\gamma_{width}$  와  $\gamma_{height}$  는 -2.5~2.5의 균등분포(uniform distribution) 난수를 사용하였으며,  $\gamma_{\dot{x}}$  와  $\gamma_{\dot{y}}$  는 각각 -6.5~6.5, -2.5~2.5의 범위를 가지는 균등분포 난수를 사용하였다.

파티클 필터에서 또 하나의 중요한 요소는 측정 모델(measurement model)이다. 측정 모델은 예측된 현재 상태를 통해 얻어진 측정 값인 likelihood를 이용하여 현재 상태를 갱신하는 단계이며, 영상을 이용한 물체 추적에서는 측정 값을 구하기 위해 다양한 특징을 사용한다. 본 논문에서는 색상 히스토그램과 HOG 서술자를 이용하여 likelihood를 계산하며, 실시간 학습 이전과 이후 두 부분을 나누어 계산 방법을 다르게 한다. 실시간 학습 이전의 likelihood를 계산하는 방법은 특징을 나타내는 두 벡터간의 유사성을 판별하는데 사용되는 Bhattacharyya coefficient를 이용하였다. 식 (10)과 식 (11)은 각각 실시간 학습을 시작하기 전까지의 색상 히스토그램과 HOG 서술자를 이용한 likelihood  $p(z|x)$  를 계산하는 방법을 나타낸다.

$$p(z|x)_{color} = \frac{1}{\sqrt{2\pi\sigma_{color}^2}} \exp\left(-\frac{1-D^2(h_x^{color}, h_{ref}^{color})}{2\sigma_{color}^2}\right) \quad (10)$$

$$\text{where, } D(h_x^{color}, h_{ref}^{color}) = \sqrt{1 - \sum_{i=1}^N \sqrt{h_{x,i}^{color} h_{ref,i}^{color}}}$$

$$p(z|x)_{hog} = \frac{1}{\sqrt{2\pi\sigma_{hog}^2}} \exp\left(-\frac{1-D^2(h_x^{hog}, h_{ref}^{hog})}{2\sigma_{hog}^2}\right) \quad (11)$$

$$\text{where, } D(h_x^{hog}, h_{ref}^{hog}) = \sqrt{1 - \sum_{i=1}^N \sqrt{h_{x,i}^{hog} h_{ref,i}^{hog}}}$$

식 (10)과 식 (11)에서의  $h_x^{color}$  와  $h_x^{hog}$  는 각 파티클에서 계산된 색상 히스토그램과 HOG 서술자를 각각 나타내며,  $h_{ref}^{color}$  와  $h_{ref}^{hog}$  는 물체 추적을 시작할 때 초기화하였던 영상의 ROI에서의 색상 히스토그램과 HOG 서술자를 각각 나타낸다. 또한  $N$  은 벡터로 표현한 색상 히스토그램과 HOG 서술자의 각각의 차원을 나타낸다. 실시간 학습 이전의 방법과 다르게 실시간 학습 이후에는 IPCA 방법을 통해 계산된

특징들의 주요 성분을 이용한 방법으로 likelihood를 계산하게 된다. 식 (12)와 식 (13)은 각각 실시간 학습 이후의 색상 히스토그램과 HOG 서술자를 이용한 likelihood 계산을 나타낸다.

$$p(z|x)_{color} = \frac{1}{\sqrt{2\pi\sigma_{color}^2}} \exp\left(-\frac{D(z_x^{color}, z_{reconst}^{color})}{2\sigma_{color}^2}\right)$$

$$\text{where, } z_{reconst}^{color} = U_{color} U_{color}^T z_x^{color}, \quad (12)$$

$$D(z_x^{color}, z_{reconst}^{color}) = \sqrt{\sum_{i=1}^N z_{x,i}^{color} - z_{reconst,i}^{color}}$$

$$p(z|x)_{hog} = \frac{1}{\sqrt{2\pi\sigma_{hog}^2}} \exp\left(-\frac{D(z_x^{hog}, z_{reconst}^{hog})}{2\sigma_{hog}^2}\right)$$

$$\text{where, } z_{reconst}^{hog} = U_{hog} U_{hog}^T z_x^{hog}, \quad (13)$$

$$D(z_x^{hog}, z_{reconst}^{hog}) = \sqrt{\sum_{i=1}^N z_{x,i}^{hog} - z_{reconst,i}^{hog}}$$

식 (12)와 식 (13)에서의  $z_{reconst}^{color}$  와  $z_{reconst}^{hog}$  는 현재 시간  $t$ 에서 계산된  $z_x^{color}$  와  $z_x^{hog}$  를 각각의 주요성분을 나타내는 행렬인  $U_{color}$  와  $U_{hog}$  를 이용하여 재구성(reconstruction)한 물체의 색상 히스토그램과 HOG 서술자를 나타낸다. 또한  $N$  은 각 특징이 가지는 차원의 수를 나타낸다. 또한 likelihood를 계산하기 위해 사용된 모든  $\sigma_{color}$  와  $\sigma_{hog}$  는 0.4로 정의하였다.

두 특징인 색상 히스토그램과 HOG 서술자에서의 likelihood는 파티클 필터의 각 파티클에 가중치를 부여하기 위해 적절하게 혼합된 형태로 사용되어야 한다. 식 (14)는 각 파티클에 가중치를 부여하기 위해 사용된 특징 혼합 방법을 나타낸다.

$$p(z|x) = \alpha p(z|x)_{color} + (1-\alpha) p(z|x)_{hog} \quad (14)$$

최종적으로 계산된  $p(z|x)$  는 각 파티클에서의 가중치로 부여해 주며,  $\alpha$  는 색상 히스토그램과 HOG 서술자를 통해 계산된 likelihood의 결합 가중치이다. 본 논문에서는 실험적 결과로  $\alpha = 0.1$  로 하여 알고리즘을 실험하였다. 이러한 이유는 빈번하게 발생하는 조명 변화로 인해 색상 히스토그램의 정보가 HOG 서술자에 비해 상대적으로 낮은 확신 도를 가지고 있기 때문이다.

본 논문에서 사용된 파티클 필터는 가중치에 따라 그 값이 적은 파티클들은 제거하고 값이 큰 파티클의 주변에 새로운 파티클들을 재생성 시키는 SIR기반의 방법이다. 이는 SIS 방법에서 발생할 수 있는 퇴화 현상을 방지하고 보다 효과적으로 물체를 추적할 수 있다[11]. 다음은 색상 히스토그램과 HOG 서술자를 이용하여 특징들을 혼합시키고, 파티클 필터를 기반으로 한 제안된 알고리즘의 Pseudo Code를 보여준다.



**IF t=0 THEN**  
 초기화: 균등하고 독립적인 난수 발생을 통해  $N$  개의 파티클  $x_0^{(i)}, i = [1, \dots, N]$  을 분포시키고, 각 파티클에 가중치  $x_0^{(i)}, i = [1, \dots, N]$  로 영상에 분포시킴

**ELSE THEN**  
 입력:  $\langle x_{t-1}^{(i)}, w_{t-1}^{(i)}, c_{t-1}^{(i)} \rangle, i = [1, \dots, N], z_t$

**FOR**  $i=1, \dots, N$  **DO**  
 - 파티클  $x_t^{(i)}$  선택하기  
 1. 균일 분포의 난수  $r \in [0, 1]$  생성하기  
 2.  $c_{t-1}^{(i)} \geq r$  인 가장 작은  $i$  찾기  
 3.  $x_t^{(i)} = x_{t-1}^{(i)}$   
 - 새로운 파티클  $x_t^{(i)} = Ax_{t-1}^{(i)} + Bz_t$  예측하기  
 - 가중치 계산하기  
 $w_t^{(i)} = p(z_t | x_{t-1}^{(i)}) = \alpha w_{t-1}^{(i),color} + (1-\alpha)w_{t-1}^{(i),hog}$

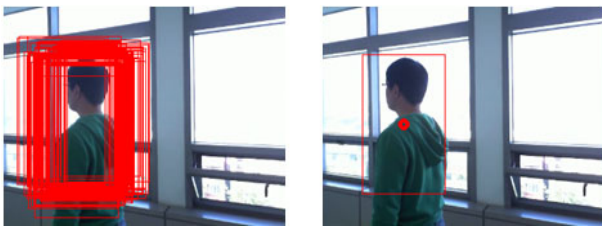
**END FOR**

가중치 정규화 하기  $w_t^{(i)} = \frac{w_t^{(i)}}{\sum_i w_t^{(i)}}$

추적 가중치 계산하기  
 $c_t^{(i)} = c_{t-1}^{(i-1)} + w_t^{(i)}, i = [1, \dots, N]$

$x_t$  의 기대 값 계산하기  $\sum_{i=1}^N w_t^{(i)} \delta(x_t - x_t^{(i)})$

**END ELSE**



(a) Process of particle filter. (b) Result of particle filter.  
 그림 6. 물체 추적을 위한 파티클 필터 과정과 결과.  
 Fig. 6. Processing and result of particle filter for object tracking.

$N$  은 파티클의 개수를 의미하며,  $N=100$  을 사용하였다.  $x_t$  은 식 (8)에서 언급되었던 각 파티클의 상태변수를 나타내며,  $w_t$  는 각 파티클의 가중치를 나타낸다. 또한  $c_t$  는 모든 파티클의 가중치를 순서대로 누적한 값을 의미한다.  $c_t$  는 pseudo code에서와 같이 난수 발생을 이용하여 확률적으로 파티클을 선택할 때 사용된다.  $x_t$  의 기대 값은 각 파티클의 상태변수  $x_t$  의 가중치  $w_t$  에 따른 가중 합을 나타내며, 이는 추적하고 있는 물체의 위치와 크기를 나타내는 것과 같다. 그림 6은 파티클 필터를 이용한 물체 추적의 결과를 나타낸다.

**VI. 실험 결과**

본 절에서는 조명 변화가 빈번히 발생하고, 주변 환경이 계속적으로 변화하며, 물체의 형태도 계속적으로 변화하는 다양한 어려움을 내포하는 환경에서 제안한 알고리즘을 실험 하였다. 표 1은 실험이 수행된 환경을 나타내며, 표 2는 실험에 사용된 대표적인 8개의 DB를 분류를 나타낸다.

표 1. 실험 환경.

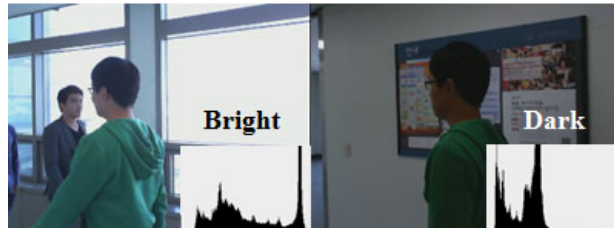
Table 1. Experimental environments.

CPU / RAM	Intel® Core™2 Quad CPU Q6600 2.4GHz / 4GB
Tool	Visual Studio 2008 / OpenCV 2.1
Camera	Microsoft Lifecam Cinema
Image resolution	640×480
Image fps	30 fps

표 2. DB 분류.

Table 2. DB classification.

환경	낮	밤	급격한 조명 변화	유사 물체 등장 (회)
DB_01	O	X	O	8
DB_02	O	X	O	6
DB_03	O	X	O	6
DB_04	O	X	O	7
DB_05	X	O	O	3
DB_06	X	O	O	3
DB_07	X	O	O	3
DB_08	X	O	O	3



(a) Rapid illumination change (Day).



(b) Rapid illumination change (Night).

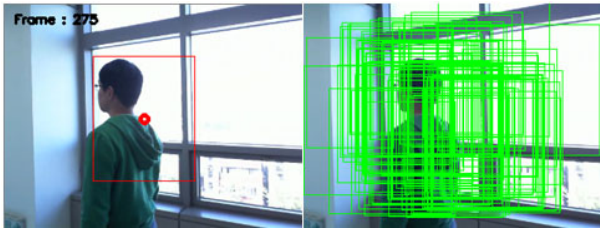
그림 7. 급격한 조명 변화의 예.

Fig. 7. Example of rapid illumination change.

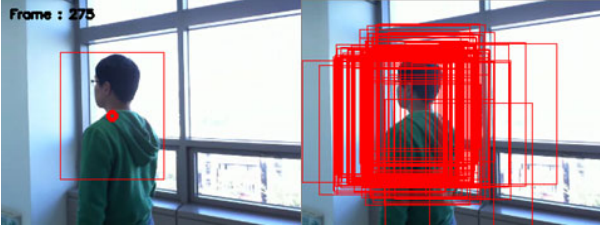
실험은 표 1과 표 2의 환경에서 수행되었으며, 8개의 DB는 동일한 시나리오로 획득하였다. 표 2의 급격한 조명 변화는 물체가 이동하는 동안 환경에서의 조명 변화가 존재하는지의 여부를 나타내며, 그림 7은 낮과 밤에 발생하는 급격한 조명 변화를 나타내었다.

유사 물체 등장은 DB안에서 물체가 이동하는 동안 유사한 물체가 영상 내에 존재하게 되는 경우를 의미한다. 제안된 알고리즘은 성능을 평가하기 위해 색상 히스토그램만을 이용한 파티클 필터, 색상 히스토그램과 HOG 서술자를 혼합한 특징을 이용한 파티클 필터와 각각 비교하였다[14,20].

색상 히스토그램을 이용한 파티클 필터는 물체의 특징을 색상만으로 고려하여 물체를 추적하기 때문에 실험에서 사용된 DB의 환경과 같이 조명 변화가 빈번히 발생하는 환경에서는 기존의 초기화된 값을 기반으로 추적하기에는 정확



(a) Result of particle filter using color histogram.



(b) Result of proposed algorithm.

그림 8. 색상 히스토그램을 이용한 파티클 필터와 제안된 알고리즘의 비교.

Fig. 8. Comparison between particle filter using color histogram and proposed algorithm.

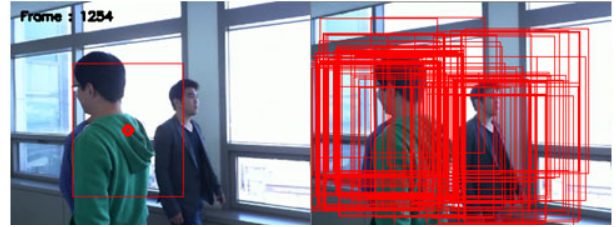
성이 떨어지게 된다. 그림 8은 색상 히스토그램만을 이용한 파티클 필터와 제안된 알고리즘의 결과를 나타낸다.

두 알고리즘의 초기화는 동일한 조건의 물체 ROI를 입력받아 물체를 추적하게 하였으며, 그림 8(a)는 색상 히스토그램을 이용한 파티클 필터의 결과를 나타낸다. 초기 ROI에서 입력된 색상 히스토그램의 측정 값을 기반으로 물체를 추적하기 때문에 초기화가 된 시점과 다른 조명 상태를 갖는 환경에서는 그림 8(a)의 오른쪽의 결과와 같이 많은 파티클이 물체에 수렴하지 않고 분산되어 있음을 알 수 있다. 분산된 파티클의 기대 값은 물체의 현 상태를 나타내기 때문에 그림 8(a)의 왼쪽의 결과와 같이 실제 물체의 위치보다 어긋난 곳에서 물체가 있음을 보여준다. 그림 8(b)의 결과와 같이 제안된 알고리즘의 결과는 색상 히스토그램만 이용한 파티클 필터의 결과보다 안정적인 물체 추적을 보여준다. 그림 8(b)의 오른쪽 결과와 같이 다수의 파티클들이 물체에 수렴하고 있으며, 이를 통해 계산된 기대 값도 그림 8(b)의 왼쪽 결과와 같이 물체를 안정적으로 표현하고 있음을 알 수 있다.

그림 9은 색상 히스토그램과 HOG 서술자를 혼합한 특징을 사용하는 파티클 필터의 결과와 제안된 알고리즘의 결과를 나타낸다. 두 알고리즘 역시 동일한 조건하에 물체를 초기화하여 추적을 수행시켰다. 그림 9(a)는 색상 히스토그램과 HOG 서술자를 혼합한 특징을 사용한 파티클 필터의 결과이다. 그림 9(a)의 오른쪽 결과와 같이 유사한 물체가 등장할시에 파티클이 유사한 물체로 많이 분포됨을 알 수 있다. 이러한 결과로 인해 파티클의 기대 값을 나타내는 왼쪽 결과와 같이 추적하는 물체와 유사한 물체 사이에서 물체를 추적하게 되는 경우를 확인할 수 있다. 그림 9(b)는 제안된 알고리즘의 결과로 그림 9(a)와 조금 다른 결과를 보인다. 학습을 이용한 제안된 알고리즘은 물체의 형태와 색상 정보의 주요 성분을 포함하고 있기 때문에 학습을 하지 않는 그림 9(a)보다 좋은 성능으로 물체를 추적하게 된다. 그림 9(b)의 오른쪽



(a) Result of particle filter using color histogram and HOG descriptor.



(b) Result of proposed algorithm.

그림 9. 색상 히스토그램과 HOG 서술자를 혼합하여 이용한 파티클 필터와 제안된 알고리즘의 비교.

Fig. 9. Comparison between particle filter using color histogram and HOG descriptor and proposed algorithm.

결과를 보면 추적하는 물체에 보다 많은 파티클이 수렴되고 있음을 확인할 수 있다. 그림 10와 표 3은 8개의 대표적인 DB에서 각각의 알고리즘을 정량적으로 평가하고 이를 표현하였다.

그림 10와 표 3에서는 색상 히스토그램을 이용한 파티클 필터를 Color로 표기하였고, 색상 히스토그램과 HOG 서술자를 혼합한 특징을 이용한 파티클 필터는 Color+HOG로 표현하였다. 또한 Frame Pos는 각 DB에서 추적에 성공한 총 프레임 수를 나타낸다. 표 3을 살펴보면 제안된 알고리즘은 다른 알고리즘에 비해 여러 DB에서 높은 성능을 보였다. 하지만 추적되는 물체의 특징을 여과 없이 누적하는 과정에서 몇 개의 DB에서는 색상 히스토그램과 HOG 서술자를 혼합한 알고리즘보다 다소 낮은 성능을 보였다. 또한, 밤에 획득된 DB에서는 충분하지 않는 조명의 효과로 인해 모든 알고리즘에서 평균적으로 낮은 성능을 보였다. 그림 11는 낮에 획득된 DB와 밤에 획득된 DB에서의 제안된 알고리즘의 결과를 보여준다.

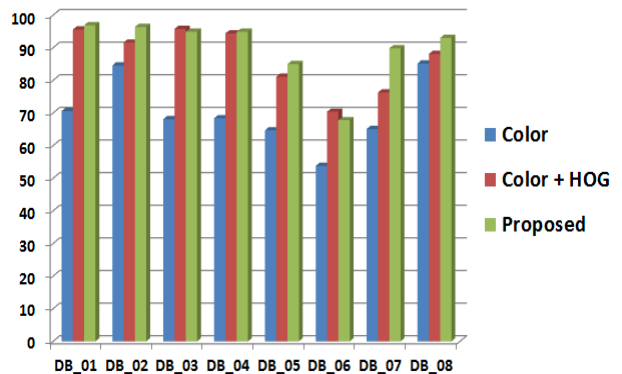


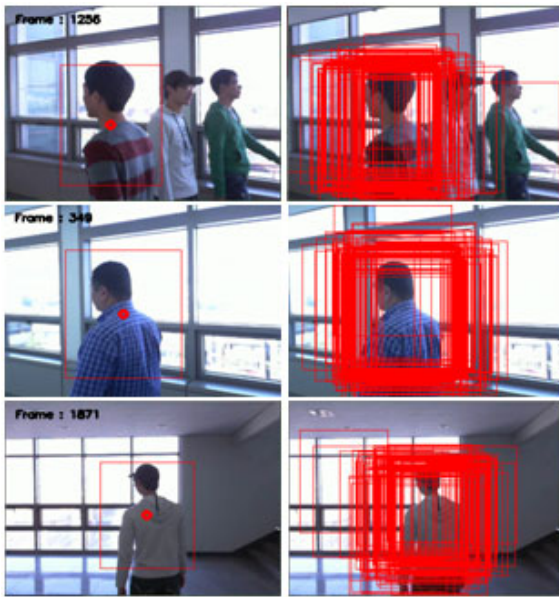
그림 10. 각 알고리즘의 성능 평가 그래프.

Fig. 10. Performance evaluation graph of each algorithm.

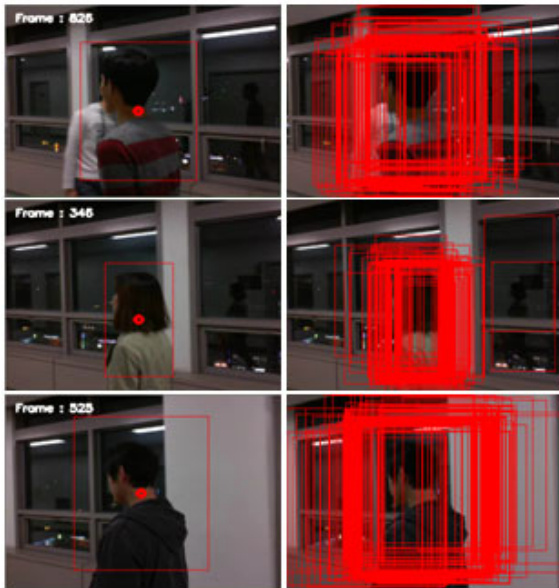
표 3. 8개의 DB에서의 알고리즘 성능 평가.

Table 3. Performance evaluation in 8 DBs.

Name	Total Frame	Color [20]		Color + HOG [14]		Proposed	
		Frame Pos.	%	Frame Pos.	%	Frame Pos.	%
DB 01	3015	2128	70.6	2883	95.6	2923	96.9
DB 02	2790	2361	84.6	2555	91.6	2689	96.4
DB 03	3545	2413	68.1	3397	95.8	3365	94.9
DB 04	3215	2198	68.4	3035	94.4	3052	94.9
DB 05	774	501	64.7	628	81.1	658	85.0
DB 06	1360	731	53.8	957	70.4	922	67.8
DB 07	1129	735	65.1	861	76.3	1014	89.8
DB 08	1347	1147	85.2	1187	88.1	1253	93.0
Average		71.1 %		90.3 %		92.4 %	
Processing Time		7 ms		13 ms		55 ms	



(a) Result of object tracking in day DBs.



(b) Result of object tracking in night DBs.

그림 11. 낮/밤 DB에서의 제안된 알고리즘의 결과.

Fig. 11. Results of proposed algorithm in day/night DBs.

## VII. 결론 및 향후 과제

본 논문에서는 이동 로봇에서 측면 카메라를 이용하여 물체를 추적하는 알고리즘을 제안하였다. 전방 카메라보다 빠른 주변 배경의 변화와 조명의 변화, 물체의 형태의 변화 등의 문제점을 극복하기 위해 물체의 색상 히스토그램과 HOG 서술자를 혼합하여 물체를 효율적으로 모델링 하였고, 파티클 필터를 통해 비선형적인 물체의 움직임을 강인하게 추적하였다. 또한, 추적되는 물체의 정보를 매 프레임 동안 누적하고 실시간 학습을 통해 주요 성분을 분석하고 이를 이용하여 형태 및 색상이 변화하는 물체를 보다 효율적으로 추적하도록 하였다. 표 3의 결과와 같이 기존의 방법보다 높은 성능을 나타냄을 확인하였다.

하지만 실험 결과에서 보이는 것과 같이 물체의 특징이 기존의 특징과 유사성을 지니는지의 판별을 하지 않고 누적한 결과로 인해 추적이 잠시 동안 실패함에 따라 성능이 저하됨을 보였다. 또한 유사 물체의 등장 시에도 기존의 색상 히스토그램과 HOG 서술자를 이용한 방법보다 추적하는 물체에 보다 많은 파티클이 수렴함으로써 상대적으로 높은 성능을 보였지만 기존의 방법과 유사하게 불안정한 성능을 보였다.

향후 이러한 문제점을 개선하기 위해 특징의 차원을 높임과 더불어 효율적으로 특징을 정의하고, 환경의 변화에 적응적인 방법으로 알고리즘이 개선이 된다면 보다 높고 안정적인 성능을 가지게 될 것이라 예상된다. 또한 제안된 알고리즘은 평균 55 ms의 처리 속도를 가지므로 CPU기반 혹은 GPU기반의 병렬 처리 프로그래밍을 적용한다면 보다 향상된 처리 시간을 통해 실시간 처리가 가능할 것으로 예상된다.

## 참고문헌

- [1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, Dec. 2006.
- [2] T. N. Hieu and W. M. S. Arnold, "Fast occluded object tracking by a robust appearance filter," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1099-1104, Aug. 2004.
- [3] W. Zuo and D. Zhang, "Bidirectional PCA with assembled matrix distance metric for image recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 36, no. 4, pp. 863-872, Aug. 2006.
- [4] F. Xu and M. Gao, "Human detection and tracking based on HOG and particle filter," *International Congress on Image and Signal Processing*, pp. 1503-1507, Oct. 2010.
- [5] G. R. Bradski and S. Clara, "Computer vision face tracking for use in a perceptual user interface," *Intel Journal*, 1998.
- [6] D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," *International Conference on Image Processing*, vol. 3, pp. 70-73, 2000.
- [7] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using camshift algorithm and multiple quantized feature spaces," *Proceedings of Pan-Sydney area workshop on Visual Information Processing*, pp. 3-7, 2004.
- [8] J. A. Corrales, P. Gil, F. A. Candelas, and F. Torres, "Tracking based on hue-saturation features with a miniaturized active vision system," *Proc. of the 40th International Symposium on Robotics*, pp. 107-112, 2009.
- [9] Q. Xuena and L. Qiang, "Target tracking and localization of



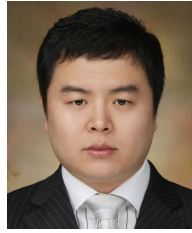
binocular mobile robot using CAMShift and SIFT,” *Proc. of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pp. 483-487, 2009.

- [10] J. H. Kim, C. H. Park, and I.-S. Kweon, “Visual tracking for non-rigid objects using rao-blackwellized particle filter,” *IEEE International Conference on Robotics and Automation*, pp. 4537-4544, May 2010.
- [11] Y. Salih and A. S. Malik, “Comparison of stochastic filtering methods for 3D tracking,” *Pattern Recognition*, vol. 44, no. 10, pp. 2711-2737, Oct. 2011.
- [12] S. Avian, “Ensemble tracking,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261-271, 2007.
- [13] D. A. Klein, D. Schulz, S. Fintrop, and A. B. Cremers, “Adaptive real-time video-tracking for arbitrary objects,” *IEEE International Conference on Intelligent Robots and Systems (ICROS)*, pp. 772-777, Oct. 2010.
- [14] L. Jin, J. Cheng, and H. Huang, “Human tracking in the complicated background by particle filter using color-histogram and hog,” *International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 1-4, Dec. 2010.
- [15] D. A. Ross, J. W. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125-141, 2007.
- [16] H. Yang, Z. Song, and R. Chen, “An incremental PCA-HOG descriptor for robust visual hand tracking,” *Lecture Notes in Computer Science*, vol. 6454, pp. 687-695, 2010.
- [17] W.-L. Lu and J. J. Little, “Simultaneous tracking and action recognition using the PCA-HOG descriptor,” *Canadian Conference on Computer and Robot Vision*, pp. 6-13, 2006.
- [18] B. Jung and G. S. Sukhatme, “Real-time motion tracking from a mobile robot,” *International Journal of Social Robotics*, vol. 2, no. 1, pp. 63-78, 2010.
- [19] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, “Robust tracking-by-detection using a detector confidence particle filter,” *IEEE International Conference on Computer Vision*, pp. 1515-1522, Oct. 2009.
- [20] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” *Proc. of the 7th European Conference on Computer Vision*, pp. 661-675, 2002.
- [21] J.-H. Kim, D.-W. Lee, K.-R. Cho, S.-Y. Jo, J.-H. Kim, and D.-I. Han, “Vision based estimation of 3-D position of target for target following guidance/control of UAV,” *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 14, no. 12, pp. 1205-1211, 2008.
- [22] J. H. Park, W. Bhan, T.-Y. Choi, H. I. Kwon, D.-I. Cho, and K. S. Kim, “Recognition performance of vestibular-ocular reflex based vision tracking system for mobile robot,” *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 15, no. 5, pp. 496-504, 2009.



**이형호**

2010년 인하대학교 정보통신 공학부(공학사). 2012년 인하대학교 정보 공학과(공학석사). 2012년~현재 LIG Nex1 전자전 연구센터 연구원. 관심분야는 패턴인식, 물체 추적, 로봇 비전, 병렬 영상처리.



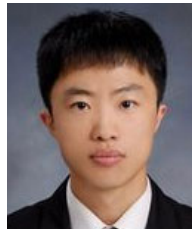
**최학남**

2007년 상명대학교 컴퓨터학과(공학석사). 2011년 인하대학교 정보 공학과(공학박사). 2011년~현재 인하대학교 슈퍼지능연구소 Post Doc.. 관심분야는 로봇비전, 의료영상처리, 머신비전, 병렬 영상처리.



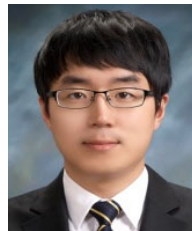
**김형래**

2011년 목원대학교 지능로봇공학과 졸업. 2011년~현재 인하대학교 로봇공학전공 석사과정 재학중. 관심분야는 로봇비전, 자동차비전.



**마승완**

2010년 인하대학교 컴퓨터정보공학과 졸업. 2010년~현재 인하대학교 로봇공학과 석사과정 재학중. 관심분야는 로봇비전, 물체인식, SLAM.



**이재홍**

2005년~현재 인하대학교 재학중. 2012년 졸업예정 및 인하대학교 정보통신공학과 석사과정 입학예정. 관심분야는 로봇비전, 물체인식 및 추적.



**김학일**

1983년 서울대학교 제어계측 공학과(공학사). 1985년 Purdue Univ. 전기/컴퓨터 공학과(공학석사). 1990년 Purdue Univ. 전기/컴퓨터 공학과(공학박사). 1990년~2001년 인하대학교 자동화 공학과 조교수. 2001년~현재 인하대학교 정보통신 공학과 교수. 2001년~현재 한국바이오인식포럼 부의장. 2001년~현재 정보보호학회 바이오인증연구회 회장. 2003년~현재 ISO/IEC JTC1/SC37 (Biometrics) 국내 및 국제 전문위원. 2004년~2008년 Elsevier Pattern Recognition Letters Associate Editor. 2005년~현재 ITU-T/SG17/WP2/Q.9 (Telebiometrics) Rapporteur. 관심분야는 패턴인식, 컴퓨터비전, 바이오인식, 초분광영상처리, 로봇비전, 의료영상처리.