

가우시안 입자 군집 최적화를 이용한 사람의 통합된 검출 및 추적

Unified Detection and Tracking of Humans Using Gaussian Particle Swarm Optimization

안 성 태*, 김 정 중, 이 주 장
(Sung-Tae An¹, Jeong-Jung Kim¹, and Ju-Jang Lee¹)
¹KAIST

Abstract: Human detection is a challenging task in many fields because it is difficult to detect humans due to their variable appearance and posture. Furthermore, it is also hard to track the detected human because of their dynamic and unpredictable behavior. The evaluation speed of method is also important as well as its accuracy. In this paper, we propose unified detection and tracking method for humans using Gaussian-PSO (Gaussian Particle Swarm Optimization) with the HOG (Histograms of Oriented Gradients) features to achieve a fast and accurate performance. Keeping the robustness of HOG features on human detection, we raise the process speed in detection and tracking so that it can be used for real-time applications. These advantages are given by a simple process which needs just one linear-SVM classifier with HOG features and Gaussian-PSO procedure for the both of detection and tracking.

Keywords: human detection, human tracking, HOG (Histograms of Oriented Gradients), PSO (Particle Swarm Optimization), gaussian-PSO

I. 서론

사람의 검출 및 추적은 지능형 차량, 영상 감시 시스템, 서비스 로봇 등에서 중요한 작업 중 하나이다[1,2]. 그리고 사람은 다양한 걸모습과 자세로 인하여 검출하기가 쉽지 않다. 검출 후 그 사람에 대한 추적 역시 그들의 동적이고 예측할 수 없는 다음 행동 때문에 결코 쉽지 않은 작업이다. 또한, 실시간 어플리케이션에서는 높은 정확도를 기반으로 빠른 검출 및 추적 결과를 얻는 것이 중요하다.

Dalal과 Triggs는 처음으로 HOG (Histograms of Oriented Gradients)를 제안하였다[3]. HOG는 다른 특징정보들에 비해 사람의 검출에 대한 뛰어난 성능으로 인하여 가장 널리 사용되는 특징정보이다. Dalal과 Triggs는 영상 내에서 사람을 검출하기 위해서 촘촘한 간격의 슬라이딩 윈도우 방식을 사용하였고, 검출 비율에 있어서 매우 뛰어난 결과를 보여주었다. 이집게도, 촘촘한 간격의 슬라이딩 윈도우 방식으로 이미지 전체를 검색하는 것은 시간 소비가 크기 때문에, 실시간 어플리케이션에 적용하기에는 속도가 충분하지 않다. Zhu는 속도를 높이기 위해 AdaBoost 알고리즘을 이용한 cascade-of-rejectors를 구성하였고, 실시간에 근접한 검출 성능을 보여준다[4]. 그러나 그들이 제안하는 방법은 분류

기를 학습하는 과정에 매우 많은 시간을 소비한다. 반면에 본 논문에서 제안하는 방법은 단 하나의 선형 SVM 분류기만을 학습하면서도 실시간으로 동작할 수 있다.

비모수적 특징 공간 분석 기술 중 하나인 평균 이동(mean shift)은 색상 히스토그램을 함께 이용해서 물체 추적에 자주 사용된다[5,6]. 그러나 색상 히스토그램 기반의 평균 이동 추적 알고리즘은 영상의 배경 환경에 많은 영향을 받는다.

파티클 필터 또한 상태 추정이나 추적에 있어서 많이 사용되는 방법이다. 따라서 영상 내 사람 추적을 비롯한 많은 응용분야에서 사용된다[7-9]. 그러나 파티클 필터를 사용한 물체 추적 방법은 차선의 지역 솔루션으로 수렴할 위험이 다소 있다.

입자 군집 최적화(PSO: Particle Swarm Optimization)는 연속 비선형 함수 최적화[10], 무효 전력 및 전압 제어[11], 전력 시스템을 위한 제어기의 모수 조절[12], 그리고 PID 제어기 설계[13] 등에 성공적으로 적용되었다. 영상 내에서

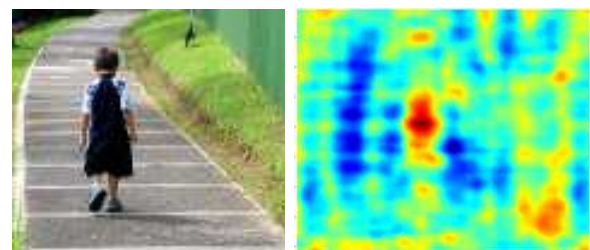


그림 1. 사람을 포함한 영상의 선형 SVM 분류기의 출력값.
Fig. 1. Map of HOG-SVM output.

* 책임저자(Corresponding Author)

논문접수: 2011. 12. 18., 수정: 2011. 12. 23., 채택확정: 2012. 1. 10.

안성태: KAIST 항공우주공학전공(ast0414@kaist.ac.kr)

김정중, 이주장: KAIST 전기 및 전자공학과

(rightcore@kaist.ac.kr/jjlee@ee.kaist.ac.kr)

※ 본 논문은 2011년 ICROS 대전충청지부 학술대회에서 초안이 발표되었음.

사람의 위치를 찾는 것 역시 비선형 최적화 문제로 생각할 수 있기 때문에, 입자 군집 최적화는 영상 내 사람 검출에도 효과적으로 적용할 수 있다[14]. 그림 1은 HOG 특징정보로 학습된 선형 SVM 분류기의 사람이 포함된 영상 입력에 대한 출력값을 보여준다.

더욱이 단일 영상이 아닌 연속적인 프레임을 다룬다면 이것들의 시간적 연속성을 활용할 수 있다. 그래서 입자 군집 최적화의 전체 과정을 나누어서 각각의 프레임마다 계산과정의 일부 단계만을 수행하도록 할 수 있다.

본 논문의 남은 부분은 다음과 같이 구성된다. 먼저 우리가 제안하는 검출 및 추적의 통합적인 방법의 전체 구조를 간략하게 설명한 후, HOG, 입자 군집 최적화 및 제안하는 방법의 각 단계들에 대해 조금 더 자세히 살펴본다. 그 다음으로 제안하는 방법 및 기존 방법들의 실험 결과를 비교해 보고 결론을 맺는다.

II. 제안하는 방법의 개요

먼저 제안하는 검출 및 추적의 통합적인 방법에 대해 간략히 살펴본다. 먼저 연속적인 비디오 프레임에서 사람을 검출 및 추적하기 위해 입자 군집 최적화 중에서도 가우시안 입자 군집 최적화(GPSO: Gaussian Particle Swarm Optimization [15])를 사용한다. 각 입자는 위치와 스케일을 상태 벡터로 가지고, 초기에 이미지 전체 영역에 균일하게 분포된다. 각 입자의 위치에서 원래의 스케일에 대해 64x128의 크기로 리사이징한 검출 윈도우로부터 HOG 특징정보를 추출한다. 추출한 HOG 특징정보를 선형 SVM 분류기를 사용해서 분류한 다음 그 출력값을 입자의 fitness 값으로 사용해서 GPSO는 최대의 fitness를 가지는 입자를 찾는다. 매 프레임에서 각 입자는 전역 최적값을 향해서 속도와 위치를 바꿔 나간다. 위 과정을 몇 번 반복한 후에, 입자들은 사람이 있는 위치로 수렴하게 된다. 일단 사람이 검출되면, 입자들이 이미지 전체 영역이 아닌 사람이 검출된 영역 주변으로 분포되는 것을 제외하고는 검출을 위한 과정과 비슷하게 추적 과정을 수행한다. 각 과정에 대한 자세한 설명은 다음에 이어진다.

III. HISTOGRAMS OF ORIENTED GRADIENTS

본 논문에서는 사람 검출을 위한 선형 SVM 분류기를 학습 및 사용하기 위해서 HOG 특징 정보를 사용한다. HOG 특징 정보는 64x128 픽셀 사이즈의 검출 윈도우로부터 추출된다. 각 검출 윈도우는 8x8 픽셀 사이즈의 셀로 나누어지고 각 2x2 셀의 그룹이 서로 겹치는 블록을 구성한다.

먼저, 단순 1-D [-1, 0, 1] 그래디언트 필터를 가우시안 스무딩 없이 사용하여서 검출 윈도우 영역에 대한 그래디언트를 계산한다. 컬러 이미지는 각 채널에대해 따로 계산한 후에 가장 큰 L2-norm을 가지는 채널의 그래디언트 벡터를 사용한다.

다음 단계로 그래디언트 각도에 대한 히스토그램을 만든다. 각 히스토그램은 각 셀 단위에 대해서 계산하며 각 픽셀의 그래디언트 각도를 0~180도의 부호 구분 없는 각도로 균등하게 9개의 구간으로 투표 한다. 각 투표는 그래디언트

크기에 의해 가중되고, 각도와 위치에 대해 인접 구간들의 중심점들 사이에서 보간법을 적용한다.

최종적으로 HOG 특징 벡터는 7x15 개의 블록으로 이루어지고, 각 블록은 2x2 개의 셀이 슬라이딩 방식으로 이루어진다. 따라서 각 블록은 서로 겹치게 되며, 한 블록 당 36-D의 특징 정보(4셀의 9bins 히스토그램)를 가지고 L2-norm을 사용해서 각각 정규화 된다. 따라서 각 64x128 픽셀의 검출 윈도우는 3780개의 값으로 이루어진 특징정보로 나타낼 수 있다. 이 특징정보를 이용해서 선형 SVM 분류기를 학습시키고, 영상을 분류하게 된다.

IV. 입자 군집 최적화

1. 표준형 입자 군집 최적화

각 입자는 자신의 위치와 속도를 상태변수로 가지며 탐색 공간 안에서 임의로 초기화 된다. 군집 내 전체 입자 개수가 N이고, 각 입자가 D 차원의 벡터일 때, t번째 반복에서의 입자들의 위치와 속도는 (1)과 (2)로 표현된다.

$$\vec{X}(t) = \{\vec{x}_1^D, \vec{x}_2^D, \dots, \vec{x}_{N-1}^D, \vec{x}_N^D\} \quad (1)$$

$$\vec{V}(t) = \{\vec{v}_1^D, \vec{v}_2^D, \dots, \vec{v}_{N-1}^D, \vec{v}_N^D\} \quad (2)$$

여기에서

$$\vec{x}_n^D(t) = \{x_n^1(t), x_n^2(t), \dots, x_n^{D-1}(t), x_n^D(t)\} \quad (3)$$

$$\vec{v}_n^D(t) = \{v_n^1(t), v_n^2(t), \dots, v_n^{D-1}(t), v_n^D(t)\} \quad (4)$$

반복이 진행될수록 입자들은 그들의 이전 최적 위치들을 보존하고 공유하면서 솔루션에 도달한다. 입자들은 최적화 과정 동안 그들의 최적 위치들을 저장하고 (5)와 (6)에 따라서 속도와 위치를 갱신한다.

$$\vec{V}(t+1) = \vec{V}(t) + \varphi_1(\vec{p}_{best} - \vec{X}(t)) + \varphi_2(\vec{l}_{best} - \vec{X}(t)) \quad (5)$$

$$\vec{X}(t+1) = \vec{X}(t) + \vec{V}(t+1) \quad (6)$$

여기서 \vec{p}_{best} 는 이전 최적 위치이고 \vec{l}_{best} 는 입자들이 지금까지 얻었던 지역 최적 위치이다. φ_1 과 φ_2 는 $\varphi_1 = rand(0, c_1)$, $\varphi_2 = rand(0, c_2)$, 여기서 c_1 은 인지적 학습 인자, c_2 는 사회적 학습 인자이다.

2. 변형 입자 군집 최적화

원래의 속도 갱신 식 (5)는 관성 가중치나 \vec{g}_{best} 를 포함하는 형태로 (7)과 (8)과 같이 확장할 수 있다.

$$\vec{V}(t+1) = \alpha \vec{V}(t) + \varphi_1(\vec{p}_{best} - \vec{X}(t)) + \varphi_2(\vec{l}_{best} - \vec{X}(t)) \quad (7)$$

$$\vec{V}(t+1) = \vec{V}(t) + \varphi_1(\vec{p}_{best} - \vec{X}(t)) + \varphi_2(\vec{g}_{best} - \vec{X}(t)) \quad (8)$$

여기서 α 는 관성계수이고 \vec{g}_{best} 는 전체 군집에서 지금까지의 최적 위치이다. (7)에 기반을 두는 변형 입자 군집 최적화는 전역 탐색과 지역 탐색의 균형을 맞추기 위해 제안되었고, (8)에 기반을 두는 변형은 수렴 속도를 가속하기 위해서 제안되었다[16,17].

R. A. Krohling은 가우시안 분포 난수 생성기를 기반으로 하는 속도 갱신 식을 사용하는 GPSO를 제안하였다[15].

GPSO는 cognition learning factor c_1 과 social learning factor c_2 를 필요로 하지 않고 표준형 PSO보다 빠르게 수렴한다. GPSO의 속도 갱신 식은 다음과 같다.

$$\vec{V}(t+1) = |\text{randn}|(\overrightarrow{p_{best}} - \vec{X}(t)) + |\text{randn}|(\overrightarrow{g_{best}} - \vec{X}(t)) \quad (9)$$

여기서 $|\text{randn}|$ 은, $\text{abs}[N(0,1)]$ 와 같은 가우시안 확률 분포의 절대값에 따라서 생성된 양의 난수로 이루어진다. 알고리즘 1은 초기화 후에 사람의 검출 및 추적에 공동으로 사용되는 GPSO의 한 스텝을 나타낸다.

알고리즘 1. 검출 및 추적을 위한 GPSO.

Algorithm 1. GPSO for detection and tracking.

```

1: procedure G_PSO(S)
2:   for  $i = 1, \dots, N$  do
3:     calculate velocity  $v_i$ 
4:      $x_i \leftarrow x_i + v_i$ 
5:   end for
6:   for  $i = 1, \dots, N$  do
7:     Extract HOG features of  $S_i$ 
8:     SVM output as a fitness value of  $S_i$ 
9:     Update  $p_{best}$  of  $S_i$  and find  $g_{best}$ 
10:  end for
11: end procedure
    
```

V. 통합된 검출 및 추적 알고리즘

다음으로, 제안하는 통합된 사람의 검출 및 추적 알고리즘에 대해서 상세히 설명한다. 다양한 응용 분야에서 사람의 검출만이 아니라 추적 또한 필요로 한다. 더욱이, 정확도와 더불어 수행 속도 역시 중요한 요소 중에 하나이다. 빠른 수행 속도를 얻기 위하여 우리는 검출 및 속도 두 가지 작업에 모두 같은 특징 정보 및 방법을 사용한다. 앞서 설명한 것과 같이 사람 검출에 대한 강인함으로 널리 사용되는 HOG 특징정보를 사용하고, 검출 및 추적 모두 동일한 가우시안 입자 군집 최적화 과정을 통해 이루어진다.

초기 프레임에서 입자들은 전체 이미지 영역에 균일하게 분포 된다. 각 입자는 프레임 안에서의 위치, 속도 그리고 64×128 픽셀 사이즈에 대한 상대적인 스케일정보를 가지고 있다. 각 위치에서 64×128 픽셀 사이즈로 리사이징한 검출 윈도우로부터 HOG 특징정보를 추출한 후에 그것을 선형 SVM 분류기로 계산한 출력 값을 적합도 값으로 하고 큰 적합도 값을 가질수록 좋은 입자로 고려한다. 각 입자는 앞서 설명한 GPSO 과정을 통해 속도 및 위치를 갱신하게 되는데 한 프레임에서 한 스텝의 GPSO 과정만을 계산한다. 몇 프레임의 GPSO 과정을 반복한 후에 입자들은 전역 최대 적합도 값의 위치에 수렴하게 된다. 만약 그 위치에서 사람이 검출되면 추적 단계를 시작하고, 그렇지 않으면 검출 과정을 반복하기 위해 다시 초기화한다.

추적 단계의 초기 프레임에서는 전체 이미지 영역이 아닌 사람이 검출된 영역 근처 주변에 입자들이 초기화 된다. 또한, 최근 몇 프레임 동안 추적된 위치의 평균 이동량을 속도 예측 벡터로 사용하고, 현재 프레임에서의 추적 실패는 다음 프레임에서의 위치 불확실성을 높인다고 가정하였

다. 결과적으로 입자들은, 사람이 검출되거나 마지막으로 추적된 위치에서 출발하는 속도 예측 벡터의 도착점을 평균으로 하고, 현재 추적 단계에서 마지막으로 추적된 프레임 이후 추적에 실패한 횟수에 비례하는 표준 편차를 따르는 가우시안 분포에 의해 뿌려지게 된다. 입자들이 뿌려지고 초기화된 이후의 과정은 검출 단계와 동일하게 진행된다.

알고리즘 2는 우리가 제안하는 GPSO를 이용한 통합된 사람의 검출 및 추적 방법의 의사 코드를 나타낸다.

알고리즘 2. 통합된 검출 및 추적.

Algorithm 2. Unified detection and tracking.

```

1:  $S_i = \{x_i, v_i, c\}$ :  $i$ -th particle
2:  $x_i$ : 2-Dimensional center coordinate of detection window represented by  $i$ -th particle in the image
3:  $V_i$ : Velocity vector of  $i$ -th particle
4:  $c_i$ : Scale factor of detection window relative to  $64 \times 128$ 
5:  $N$ : The number of particles
6:  $iter$ : The maximum number of iterations
7:  $PV$ : The predicted velocity of human
8: procedure SDAT( $N, iter$ )
9:    $Generation \leftarrow 0$ 
10:   $Mode \leftarrow Detection$ 
11:  Get the frame
12:  for  $i = 1, \dots, N$  do
13:    Initialize each particle  $S_i$  for detection
14:     $-x_i(0)$  uniformly distributed
15:     $-v_i(0) \leftarrow 0$ 
16:     $-c_i \leftarrow$  random scale in predefined range
17:  end for
18:  repeat
19:     $Generation \leftarrow Generation + 1$ 
20:    Get the frame
21:    if  $Mode = Detection$  then
22:      G_PSO(S)
23:      if Human is detected at  $g_{best}$  then
24:        Initialize each particle  $S_i$  for tracking
25:         $-x_i(0) \sim N(x_{g_{best}} + PV, \sigma^2)$ 
26:         $-v_i(0) \leftarrow 0$ 
27:         $-c_i \leftarrow$  random scale in predefined range
28:         $Generation \leftarrow 0$ 
29:         $Mode \leftarrow Tracking$ 
30:      else
31:        if  $Generation \geq iter$  then
32:          Initialize each particle  $S_i$  for detection
33:           $Generation \leftarrow 0$ 
34:        else
35:          Continue
36:        end if
37:      end if
38:    else  $Mode = Tracking$ 
39:      G_PSO(S)
40:      if Human is detected at  $g_{best}$  then
41:        Initialize each particle  $S_i$  for tracking
42:         $Generation \leftarrow 0$ 
43:      else
44:        if  $Generation \geq iter$  then
45:          Initialize each particle  $S_i$  for detection
46:           $Generation \leftarrow 0$ 
47:           $Mode \leftarrow Detection$ 
48:        else
49:          Continue
50:        end if
51:      end if
52:    end if
53:  until End of Frames
54: end procedure
    
```


VI. 실험 결과

먼저 실험은 Intel Core2Duo 1.86GHz CPU와 4GB의 RAM을 장착한 PC에서 Windows 7 환경 하에 진행되었다. C++ 언어와 OpenCV 라이브러리의 영상 처리 함수들을 사용하여 프로그래밍 하였다. 선형 SVM 분류기를 학습하기 위해서 INRIA Person Dataset [18]을 사용하였고, ViSOR 감시 영상 자료 [19]를 제안하는 알고리즘을 테스트하기 위해 사용하였다.

640x480 픽셀 사이즈의 영상에 대해 검출 및 추적 단계 모두 30개의 입자들이 사용하고 최대 10번의 GPSO 과정 반복으로 계산되었다. 그림 2와 3은 제안하는 통합된 검출

및 추적 알고리즘이 어떻게 동작하는지 보여준다. 먼저 30개의 입자들이 영상 전체에 뿌려진 빨간 색의 직사각형 윈도우로 나타내어지고 있다. 입자들은 다음 프레임의 좌측 상단에 있는 사람을 향해 이동한다. 대부분의 입자들이 5~7 프레임 후에 사람의 있는 위치로 수렴하였고, 최종적으로 10번 프레임에서 녹색 직사각형 윈도우로 표현된 전역 최적 위치에서 사람이 검출되었다. 추적 단계에서는 입자들이 검출된 영역 근처로 초기화되기 때문에 검출 단계보다 빠르게 전역 최적 위치로 수렴하는 것을 볼 수 있다. 또한, 제안하는 통합 알고리즘은 그림 4의 뛰어가는 장면과 같이 역동적으로 행동하는 상황에서도 잘 추적하는 것을 보여준



그림 2. 검출 단계.
Fig. 2. Detection phase.



그림 3. 추적 단계.
Fig. 3. Tracking phase.



그림 4. 역동적인 장면.
Fig. 4. Dynamic scene.

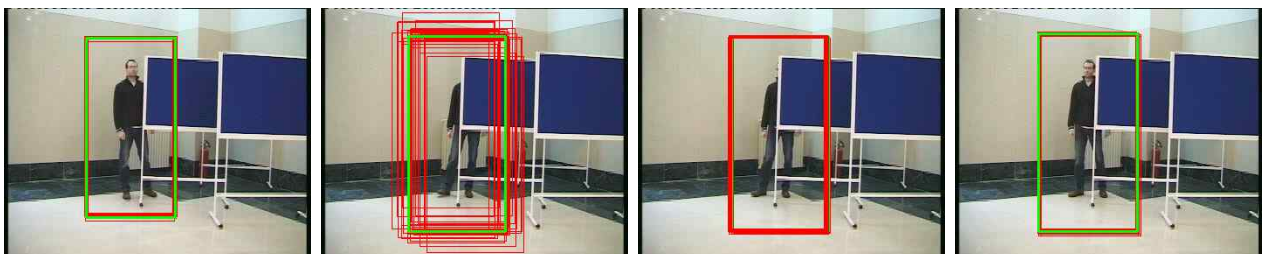


그림 5. 부분적으로 가려진 장면.
Fig. 5. Partially occluded scene.

표 1. 입자 개수에 따른 제안된 알고리즘의 실험 결과.

Table 1. The result of the proposed algorithm according to the number of particles.

Particles	Avg. # of Frames to Detect	Tracked Frames	FPS
10	60	608/729 (83.4 %)	58
20	20	757/759 (99.7 %)	24
30	15	761/761 (100 %)	15
40	13	742/766 (96.8 %)	12
50	11	760/768 (98.9 %)	10

다. 640x480 픽셀 사이즈 영상에 대한 검출과 추적 모두 부가적인 프로그램 최적화 없이 평균 초당 15프레임의 비율로 동작한다. 제안하는 통합 알고리즘이 그림 5에서와 같이 사람이 다른 물체에 의해 부분적으로 가려진 상황에 대해서도 잘 동작하는 지 실험해 보았다. 비록 사람 몸의 절반 이상이 가려지만 잠시 추적에 실패하다가, 장애물에서 벗어나는 순간 곧바로 다시 추적되는 것을 볼 수 있다. 가려지는 시간이 길어지면 계속되는 추적 실패의 반복으로 추적 단계가 종료되고 다시 검출 단계로 돌아가지만, 추적 단계의 허용 실패 횟수를 사용자가 늘릴 수도 있고, 그렇지 않더라도 검출 단계를 반복하는 과정 중에서 영상의 사람이 장애물을 벗어나면 바로 검출하여 다시 추적 단계로 넘어가게 된다.

다음으로 640x480 픽셀 사이즈의 연속 프레임 영상에서 GPSO의 총 입자 개수에 따라서 검출 및 추적의 성능이 어떻게 다른지 살펴보았다. 표 1의 첫 번째 행은 사용된 총 입자의 개수, 두 번째 행은 사람이 처음 검출되기까지 소요된 평균 프레임 수, 세 번째 행은 검출된 직후부터 남아있는 프레임들 중 처음으로 실패할 때 까지 추적된 평균 프레임 수, 그리고 마지막 행은 평균 초당 작업 프레임 수이다. 더 적은 수의 입자를 사용할수록 훨씬 빠른 평균 초당 프레임 수로 동작하지만 사람을 검출하기까지 소요되는 프레임 수는 증가하는 것을 보여준다. 640x480 픽셀 사이즈의 경우에는 30개의 입자를 사용했을 때, 검출도 어느 정도 빨리 되고, 추적도 거의 완벽하게 되어서 효율적인 입자 개수라고 할 수 있지만, 이것은 영상의 크기와 영상에 나타나는 사람의 크기에 따라서 달라질 수 있다.

VII. 결론

본 논문에서는 HOG 특징정보와 함께 가우시안 입자 군집 최적화를 사용하는 통합된 사람의 검출 및 추적 알고리즘을 제안하였다. HOG 특징정보의 사람 검출에 대한 강인함을 유지하면서도, 검출 및 추적의 동작 속도를 증가시켰고, 단일 정적 이미지 보다 연속된 영상 프레임에 적용할 때 시간적 연속 특성을 이용한 효율성을 높일 수 있다. 따라서 영상 감시 시스템, 지능형 차량, 서비스 로봇 등을 비롯한 다양한 실시간 애플리케이션에 사용될 수 있다. 또한 이러한 장점들은 HOG 특징정보로 학습한 단 한 개의 선형 SVM 분류기와 비교적 단순한 가우시안 입자 군집 최적화 과정을 검출 및 추적 모두에 통합적으로 사용해서 얻을 수 있는 것들이다.

참고문헌

[1] B.-R. Lee, Q.-B. Truong, H.-S. Kim and Y.-H. Bae, "A Study on the Pedestrian Detection on the Road Using Machine Vision," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 17, no. 5, pp. 490-498, May 2011.

[2] S.-H. Kim, I.-K. Jung, C.-W. Park and J.-H. Hwang, "Efficient Object Tracking System Using the Fusion of a CCD Camera and an Infrared Camera," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 17, no. 3, pp. 229-235, Mar. 2011.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. of 2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp.886-893, 25-25 June 2005.

[4] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," *Proc. of 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 1491- 1498, 2006.

[5] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of nonrigid objects using mean shift," *Proc. of 2000 IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 142-149, 2000.

[7] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online non-linear/non-gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[8] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," *Proc. of 2009 12th IEEE Int'l Conf. on Computer Vision*, pp. 1515-1522. Oct. 2009.

[9] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1728-1740, Oct. 2008.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. of IEEE Int'l Conf. on Neural Networks*, vol. 4, pp. 1942-1948, Nov./Dec. 1995.

[11] Y. Fukuyama and H. Yoshida, "A particle swarm optimization for reactive power and voltage control in electric power systems," *Proc. of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 87-93, 2001.

[12] T. Okada, T. Watanabe, and K. Yasuda, "Parameter tun-

ing of fixed structure controller for power system stability enhancement,” *Proc. of Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, vol. 1, pp. 162-167, Oct. 2002.

- [13] Y. Zheng, L. Ma, L. Zhang, and J. Qian, “Robust pid controller design using particle swarm optimizer,” *Proc. of 2003 IEEE International Symposium on Intelligent Control*, pp. 974-979, Oct. 2003.
- [14] S.-T. An and J.-J. Lee, “Detection and tracking of humans using particle swarm optimization,” *Proc. of 2011 ICROS (Institute of Control, Robotics and Systems) Daejeon Chungcheong Branch Conference (in Korean)*, pp. 40-44, Dec. 2011.
- [15] R. Krohling, “Gaussian swarm: a novel particle swarm optimization algorithm,” *Proc. of 2004 IEEE Conf. on Cybernetics and Intelligent Systems*, vol. 1, pp. 372-376, Dec. 2004.
- [16] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” *Proc. of 1998 IEEE International Conf. on Evolutionary Computation*, pp. 69-73, May 1998.
- [17] J. Kennedy, “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance,” *Proc. of the 1999 Congress on Evolutionary Computation*, vol. 3, pp. 3, 1999.
- [18] N. Dalal, INRIA Person Dataset, [Online], Available: <http://pascal.inrialpes.fr/data/human>
- [19] ViSOR repository surveillance video data, [Online], Available: <http://www.openvisor.org>



안 성 태

2010년 한양대학교 전자전기컴퓨터공학부 졸업. 2012년 KAIST 전기 및 전자공학과 석사. 2012년~현재 KAIST 항공우주공학전공 박사과정. 관심분야는 인공지능 및 이동로봇.



김 정 중

2006년 전북대학교 전자정보공학부 졸업. 2008년 KAIST 로봇공학학제전공 졸업. 2008년~현재 KAIST 전기 및 전자공학과 박사과정. 관심분야는 이동로봇 및 기계학습.



이 주 장

1973년 서울대학교 전기공학과 공학사. 1977년 서울대학교 전기공학과 공학석사. 1984년 Univ. of Wisconsin 전기공학과 공학박사. 1977년~1978년 Korean Electric Research & Testing Institute in Korea 연구원. 1978년~1979년 G.T.E Automatic Electric Co., U.S.A. Design and processing engineer. 1983년 Wisconsin Electric Power Co., U.S.A.의 프로젝트 엔지니어. 1984년~현재 KAIST 전기 및 전자공학과 교수. 1987년 영국 런던의 Imperial College의 로보틱스랩 방문 교수. 1991년~1992년 미국 Pittsburgh의 Carnegie-Mellon University Robotics Institute 방문 교수. 현 IEEE Fellow Member (IEEE Robotics and Automation Society, IEEE Control Society, IEEE Industrial Electronics Society, IEEE Evolutionary Computation Society). 현 KIEE, KITE, KISS, SICE and ICROS Society의 Fellow Member 및 IEEE Industrial Electronics Society의 부회장. 연구 관심분야는 인공지능을 이용한 비서/동반자 로봇, 강인제어, 지능제어.