

칼만 필터와 가변적 탐색 윈도우 기법을 적용한 강인한 이동 물체 추적 알고리즘

Robust Tracking Algorithm for Moving Object using Kalman Filter and Variable Search Window Technique

김 영 균, 현 병 용, 조 영 완, 서 기 성*
(Youngkyun Kim¹, Byeongyong Hyeon¹, Youngwan Cho¹, and Kisung Seo¹)
¹Seokyeong University

Abstract: This paper introduces robust tracking algorithm for fast and erratic moving object. CAMSHIFT algorithm has less computation and efficient performance for object tracking. However, the method fails to track a object if it moves out of search window by fast velocity and/or large movement. The size of the search window in CAMSHIFT algorithm should be selected manually also. To solve these problems, we propose an efficient prediction technique for fast movement of object using Kalman Filter with automatic initial setting and variable configuration technique for search window. The proposed method is compared to the traditional CAMSHIFT algorithm for searching and tracking performance of objects on test image frames.

Keywords: CAMSHIFT, Kalman filter, object tracking, variable sized search-window

I. 서론

대상 물체에 대한 인식 및 추적은 영상 처리의 대표적인 연구 분야로써 사람의 얼굴 및 제스처 인식을 통한 Human-Interaction, 보안 및 감시 시스템, 지능형 로봇에 이르기까지 다양한 형태로 연구되어 오고 있다[1-4]. 뿐만 아니라 하드웨어의 성능 향상과 저가의 캠 제품들로 인하여 기술의 발전이 가속화되고 있으며, 그와 더불어 다양한 알고리즘들이 다음과 같이 제안되어 왔다.

물체 추적은 움직이는 대상의 특징 정보를 추출하여 대상을 검출하고, 추출된 정보를 기반으로 특정 대상을 추적하게 된다. 대상의 윤곽선이나 코너점과 같은 특징점을 기반으로 물체를 추적하는 Condensation [5] 알고리즘이나 SIFT [6]와 같은 알고리즘은 정확성이 매우 뛰어난 반면 연산 시간이 오래 걸려서 실시간 시스템에 적용하기 어려운 문제점을 가진다. 그에 반해 대상의 컬러 정보를 이용하여 물체의 탐색 및 추적을 수행하는 Mean Shift [7]와 CAMSHIFT [8] 알고리즘은 구현의 편의와 빠른 연산시간을 통해 실시간 시스템에 적용하기 용이한 장점을 가지고 있다.

그 중에서도 CAMSHIFT(Continuously Adaptive Mean Shift)는 대상의 색상 분포를 기반으로 하여 빠르게 대상을 추적하는 알고리즘으로 최근까지도 얼굴 추적 및 로봇에 적용

되어 대상을 추적하는 등의 다양한 연구에 활용되며 강인한 추적 성능을 보여주고 있다[9,10]. 하지만 기존의 실험들에서는 물체의 움직임이 비교적 간단하고, 영상 내에서 물체가 사라지거나 나타나는 등의 급격한 변화가 발생하지 않으며, 영상 처리시간에 비해 느린 움직임을 보이는 상황들을 대상으로 하고 있다.

본 논문에서는 대상 물체가 카메라 시점을 완전히 벗어나거나 빠른 속도로 방향의 전환 및 급격한 움직임이 발생하는 물체에 대해 지속적으로 추적이 가능한 알고리즘을 제안한다. 이를 위해 기존의 CAMSHIFT 알고리즘에 물체의 움직임을 예측하기 위한 칼만 필터[11]를 적용하고, 가변적인 탐색 윈도우 설정과 자동적인 초기 탐색 윈도우 설정기법의 결합을 통해 급격한 움직임에 대처 가능하도록 설계하였다. 제안된 통합기법은 캠을 통해 촬영된 테스트 영상을 대상으로 물체 탐색 및 추적 성능에 대해서 기존의 알고리즘과 비교 실험 되었다.

II. CAMSHIFT 알고리즘

1. CAMSHIFT (Continuously Adaptive Mean Shift)

CAMSHIFT [8] 알고리즘은 특정 커널(탐색 윈도우)을 사용하여 탐색 윈도우 내 추적 색상의 분포 중심으로 수렴하며 대상을 추적하는 알고리즘이다. Mean Shift [7] 알고리즘을 기반으로 설계되었으며, Mean Shift의 단점인 고정된 커널 크기로 인한 지역 최대점으로서의 수렴 문제를 해결하기 위하여 영상의 모멘트 정보를 계산하여 색상 분포의 중심점, 크기, 회전 각도 등의 정보를 추출함으로써 커널의 크기를 수정하며 대상을 추적한다.

CAMSHIFT 알고리즘의 기본적인 동작 순서는 다음과 같다.
Step 1: 탐색 윈도우(Search Window)의 초기 위치 및 크기

* 책임저자(Corresponding Author)

논문접수: 2011. 1. 5., 수정: 2012. 2. 16., 채택확정: 2012. 5. 25.

김영균, 현병용, 서기성: 서경대학교 전자공학과

(dudrbs072@naver.com/mrhyunby@naver.com/ksseo@skuniv.ac.kr)

조영완: 서경대학교 컴퓨터공학과(ywch@skuniv.ac.kr)

※ 이 논문은 2009, 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2009-0071419, NRF-2011-0009958).

를 설정한다.

Step 2: 컬러 확률 분포를 계산하고, 탐색 윈도우의 중심을 찾기 위해 Mean Shift 알고리즘을 수행한다.

Step 3: 영상 모멘트 정보를 통해 구해진 컬러 분포의 중심 위치와 크기를 통해 탐색 윈도우를 재설정 한다.

Step 4: 재설정된 탐색 윈도우를 사용하여 Mean Shift 알고리즘을 다시 수행하고 탐색 윈도우가 수렴되거나 정해진 횟수만큼 Step 2~4 과정을 반복 수행한다.

2. CAMSHIFT 알고리즘의 문제점

CAMSHIFT 알고리즘은 최근까지도 얼굴 인식이나 대상 추적 및 다양한 영상처리 연구에 활용되고 있을 만큼 매우 효율적인 성능을 보여주고 있으나, 아직까지 다음과 같이 개선되어야 할 문제점들이 존재한다.

2.1 초기 탐색 윈도우 설정 문제

CAMSHFIT 알고리즘을 수행하기 위해서는 사용자에게 의해 초기 탐색 윈도우의 위치 및 크기를 설정해야 된다. 대상을 추적하기 위해서 특정 색상 및 영역의 설정은 불가피 하지만 알고리즘의 반복적인 수행이나 실제 시스템에 적용될 경우 사용자에게 의해 매번 탐색 윈도우를 설정하기에는 어려움이 따른다.

초기에 자동적인 탐색 윈도우를 설정하기 위하여 기존의 접근 방법들은 대부분 배경과 객체의 분리를 통해 대상을 검출함으로써 탐색 윈도우의 위치를 지정하였다[12]. 그러나 이러한 방법들은 배경에 대한 학습이나 모델링 정보를 필요로 하며, 배경이 변하거나 카메라가 움직일 경우 정확한 대상 추출의 어려움이 있다.

2.2 탐색 윈도우를 벗어난 대상 물체의 추적 문제

CAMSHIFT 알고리즘은 탐색 윈도우로 불리는 특정 커널을 기반으로 물체의 추적이 이루어지며, 탐색 윈도우내 색상 분포를 통해 지속적인 탐색이 이루어진다. 그렇기 때문에 탐색 윈도우 내 대상 물체가 존재하지 않을 경우 더 이상의 탐색이 불가능한 문제점을 가지고 있다. 특히, 빠르게 이동하는 대상 물체의 경우, 탐색 윈도우에서 물체가 벗어나 더 이상 탐색이 불가능할 수가 있다.

이러한 문제점을 해결하기 위하여 물체의 움직임에 따른 위치를 예측하기 위한 칼만 필터의 적용이나 배경 영상과의 차이를 통한 예측 모델을 생성하여 추적 성능을 향상시키기 위한 연구가 지속적으로 진행되고 있다[13,14]. 하지만 이러한 모델들을 통해 예측된 결과를 사용하더라도 아직까지 선형적인 움직임에 대해서만 적용이 이루어지고 있으며, 불규칙한 움직임을 가지는 실제 대상에 적용하기에는 어려움이 있다.

III. 개선된 CAMSHIFT 알고리즘

II 장에서 제시한 문제점을 보완하기 위하여 본 논문에서는 다음과 같은 기법들을 제안하여 CAMSHIFT 알고리즘에 결합한다. 이를 통해 설계된 전체 알고리즘의 동작 순서는 그림 1과 같다.

Step 1: 입력 영상의 역투영 영상에 대해 영역 분할에 따른 초기 탐색 윈도우를 설정한다.

Step 2: 기존의 CAMSHIFT 알고리즘을 수행하여 탐색 윈

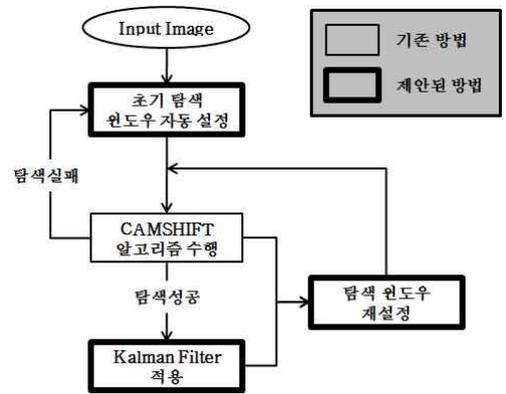


그림 1. 제안된 CAMSHIFT 기반 알고리즘.

Fig. 1. Proposed CAMSHIFT based algorithm.

도우 내 컬러분포를 계산한다.

Step 3: 계산된 컬러분포에 대한 중심 좌표와 수행시간을 칼만 필터에 적용하여 다음 상태에서의 물체의 위치를 예측하고 칼만 필터를 갱신 한다.

Step 4: 칼만 필터에 의해 예측된 탐색 윈도우와 CAMSHIFT 알고리즘에 의해 계산된 탐색 윈도우를 통해 탐색 윈도우를 재설정한다.

Step 5: 물체의 탐색이 성공하면 Step 3~5를 반복 수행한다. 물체 탐색이 실패하면 Step 2부터 다시 수행한다.

1. 분할 영역에 따른 초기 탐색 윈도우 자동 설정

초기 탐색 윈도우를 자동적으로 설정하기 위하여 그림 2와 같이 입력 영상의 역투영 영상에 대해 가로, 세로 20 픽셀 크기의 영역으로 분할하여 각 분할 영역에 대한 색상 분포를 계산한다.

색상의 분포는 검출하고자 하는 대상의 영상 파일을 입력파일로 읽어들이어 사용한다. 카메라에 의해 입력된 영상은 잡음 성분이 추가될 수 있기 때문에, 가우시안 필터를 통해 기본적인 잡음을 제거하고, 분할된 영역 내 특정 픽셀 수 이상 검출된 지역에 대해서만 탐색 윈도우를 설정하게 된다. 본 연구에서는 100 픽셀 이상 검출된 영역에 대해서 후보군으로 선정한 후 최대 검출 영역을 초기 탐색 윈도우로 설정하여 사용하였다.

2. 칼만 필터를 통한 탐색 위치 예측

2.1 칼만 필터(Kalman Filter)

칼만 필터(KF) [11]는 수학적 이론을 기반으로 하나의 선형 시스템의 시간에 따른 변화를 예측하여 발생될 수 있

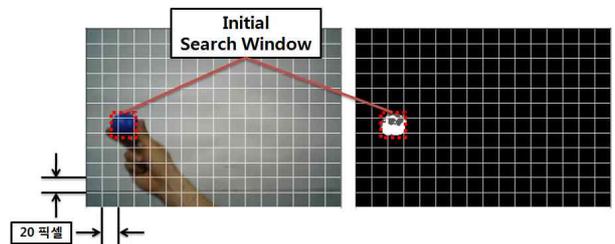


그림 2. 분할 영역에 의한 초기 탐색 윈도우 설정.

Fig. 2. Setting of initial searching window by segmentation region.

는 오류를 최소화 하는 기법이다.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ z_k &= Hx_k + v_k \end{aligned} \quad (1)$$

상태방정식 x_{k+1} 는 시스템의 전반적인 신호과형을 표현하며 프로세스 모델, 플랜트 모델 등으로 불린다. x 는 시스템의 상태를 나타내는 벡터를 의미하고 k 는 시간, u 는 시스템의 입력, w 는 프로세스 잡음을 의미한다. 측정방정식은 시스템 신호 중 측정 가능한 값들을 표현한 방정식으로 측정 모델, 센서 모델 등으로 불린다. 이때 v 는 측정 잡음을 나타낸다.

2.2 칼만 필터를 통한 탐색 위치 예측

이동 물체의 위치를 칼만 필터를 통해 예측하기 위해서는 물체 추적에 대한 상태 방정식과 측정 방정식을 설계해야 한다.

$$\begin{bmatrix} P_{x_{k+1}} \\ P_{y_{k+1}} \\ V_{x_{k+1}} \\ V_{y_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{x_k} \\ P_{y_k} \\ V_{x_k} \\ V_{y_k} \end{bmatrix} + w \quad (2)$$

물체 추적을 위하여 식 2와 같은 상태 방정식을 사용하였으며, 이때 P_{x_k} , P_{y_k} 는 이동물체의 중심좌표, V_{x_k} , V_{y_k} 는 이동물체의 속도, T 는 영상 프레임간의 시간차를 의미한다.

$$z_k = [1 \ 1 \ 0 \ 0] \begin{bmatrix} P_{x_k} \\ P_{y_k} \\ V_{x_k} \\ V_{y_k} \end{bmatrix} + v_k \quad (3)$$

측정 방정식은 식 3과 같이 정의되며, 이를 통해 이동 물체의 예측된 중심 좌표를 계산하게 된다.

3. 탐색 윈도우 재설정 기법

적용된 칼만 필터는 선형적인 시스템 방정식에 의해 설계되어 등속으로 움직이는 이동 물체에 대해서는 정확한 위치의 예측이 가능하나 불규칙적으로 움직이는 물체에 대해서는 예측이 어려운 단점을 가진다. 이를 해결하기 위해서 물체의 이동성과 칼만 필터의 예측된 위치를 고려한 가변적인 탐색 윈도우 설정 기법을 적용하였다.

그림 3은 이동하는 물체에 대한 탐색 윈도우 설정 방법을 보인다. 물체가 이동하고 있을 경우 칼만 필터에 의해 예측된 위치는 이동 방향과 속도에 따라 정해진다. 하지만 물체의 이동 속도가 변하거나 급격하게 방향이 변할 경우 예측된 위치를 벗어날 경우가 발생한다. 이를 보완하기 위하여 CAMSHIFT 알고리즘에 의해 계산된 탐색 윈도우와 칼만 필터에 의해 예측된 탐색 윈도우를 포괄하는 범위로 탐색 윈도우를 재설정하여 물체의 불규칙한 움직임에 대해 지속적인 추적이 가능하도록 설정하였다.

그림 4는 정지된 물체에 대한 탐색 윈도우 설정 방법을 보인다. 물체가 이동하고 있을 때는 칼만 필터에 의해 다음 상태의 위치가 예측가능하나 정지된 물체일 경우 예측된 위치와 CAMSHIFT에 의해 계산된 위치가 현재 물체 범위

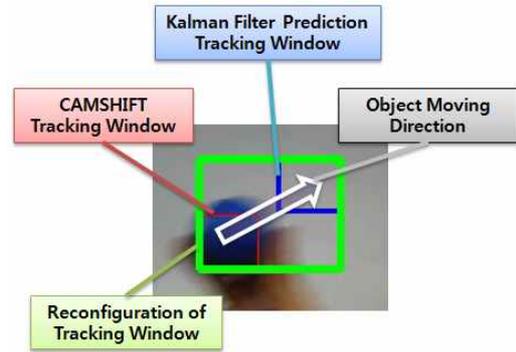


그림 3. 이동하는 물체에 대한 탐색 윈도우 설정.
Fig. 3. Setting of search windows for moving object.

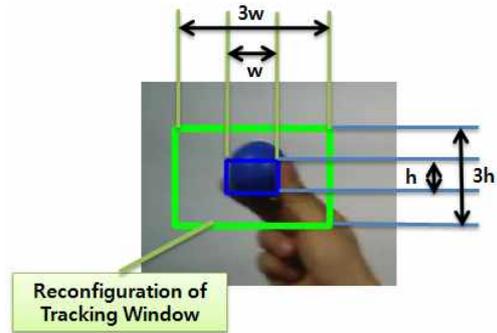


그림 4. 정지된 물체에 대한 탐색 윈도우 설정.
Fig. 4. Setting of search windows for fixed object.

로 한정되기 때문에 갑작스런 이동에 대한 추적이 어려움이 있다. 이를 해결하기 위해 물체가 정지되어 예측된 위치와 계산된 위치가 동일할 경우 현재 탐색 윈도우의 3배 크기로 범위를 확장하여 급변하는 물체에 대한 추적이 가능하도록 구성하였다.

IV. 실험 및 결과 분석

1. 실험 환경

대상 물체의 탐색 및 추적 실험을 위해 Logitech사의 QuickCam Pro for Notebooks 제품을 사용하였고, CAMSHIFT 알고리즘은 Intel에서 제공하는 영상처리 라이브러리인 OpenCV [15]를 사용하였다. 동일 조건에서의 실험을 위해 캠을 통해 영상을 촬영하여 동영상 파일로 저장한 후 대상 물체의 탐색 및 추적 실험을 수행하였으며, 실험에 사용된 PC는 Core2Duo 2.66Ghz, 4GB RAM의 사양을 가진다. 실험을 위해 사용된 물체는 지름 5cm의 플라스틱 공이다.

2. 자동적인 초기 탐색 윈도우 설정 실험

제안된 기법의 검증을 위하여 일정 높이에서 떨어트린 공에 대해 자동적으로 탐색 윈도우를 설정하여 빠른 속도로 이동하는 대상 물체의 인식 및 지속적인 추적이 이루어짐을 확인하였다.

3. 가변적인 탐색 윈도우 설정 실험

제안된 탐색 윈도우 설정 기법의 정확도를 확인하기 위하여 간단한 샘플 영상에서 각각 설정된 윈도우에 대한 결과를 확인하였다.

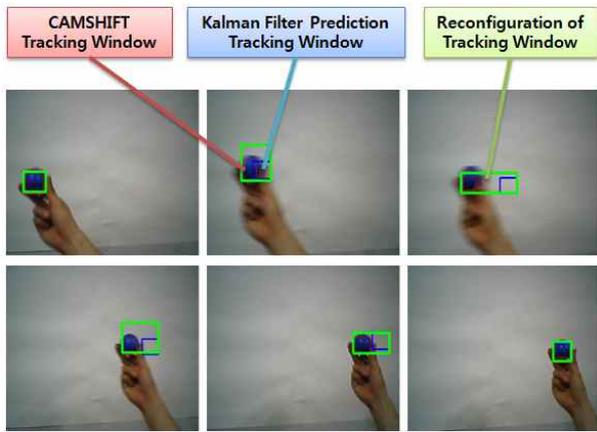


그림 5. 가변적인 탐색 윈도우의 결과 영상.

Fig. 5. Resulting images of variable search window.

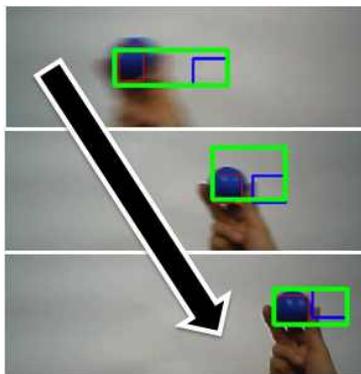


그림 6. 물체 이동에 따른 탐색 윈도우의 변화.

Fig. 6. Change of search window for movements of object.

그림 5는 이동하는 물체에 대해 CAMSHIFT에 의해 계산된 탐색 윈도우와 칼만 필터에 의해 예측된 탐색 윈도우, 그리고 이를 통해 재설정된 탐색 윈도우 영역을 나타낸다.

CAMSHIFT 알고리즘에 의해 계산된 탐색 윈도우는 색상 분포의 중심 영역에 대해 설정되므로, 영상에서 현재 물체 위치에 설정된 것을 알 수 있다. 칼만 필터에 의해 예측된 탐색 윈도우는 물체의 속도와 이동방향에 따라 설정되므로, 속도가 빨라질수록 더 멀리 떨어진 지점에 설정된 것을 확인할 수 있다. 이렇게 설정된 두 개의 탐색 윈도우를 포함하여 재설정된 탐색 윈도우의 영역을 확인할 수 있다.

그림 6에서 볼 수 있듯이 기존의 CAMSHIFT 알고리즘에서 계산된 탐색 윈도우만을 적용할 경우 물체의 빠른 이동 속도에 의해 다음 프레임에서 탐색 윈도우내 물체가 존재하지 않으며, 더 이상의 추적이 불가능하다. 그러나 제안된 방법에서는 칼만 필터를 통해 예측된 위치까지 탐색 윈도우를 확장함으로써 다음 프레임에서의 물체 위치가 재설정된 탐색 윈도우내에 위치하여 지속적인 추적이 가능하였다.

4. 물체 탐색 및 추적 실험

실제적인 물체 탐색 및 추적 실험을 위해 약 500프레임 이상 촬영된 영상을 통해 10프레임 간격으로 기존의 CAMSHIFT와 칼만 필터가 적용된 개선된 CAMSHIFT 알고리즘에 대해 비교 실험 하였다.

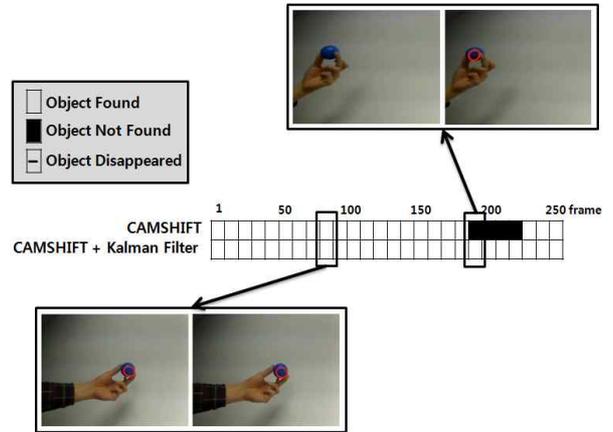


그림 7. 물체의 움직임이 적은 영상에서의 실험 결과.

Fig. 7. Experiment results with less movement of object.

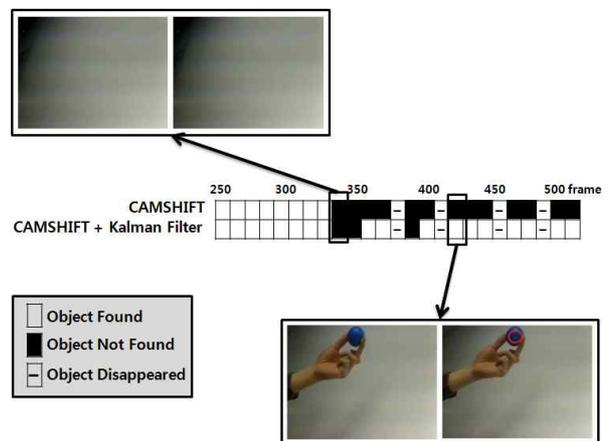


그림 8. 물체의 움직임이 큰 영상에서의 실험 결과.

Fig. 8. Experiment results with large movement of object.

4.1 물체의 움직임이 적은 영상에서의 실험

그림 7은 1~250 프레임까지의 측정된 결과를 나타낸다. 프레임 구간에서 흰색 부분은 정확한 탐색과 추적이 이루어진 부분이고, 검은색 부분은 물체의 탐색이 수행되지 못한 프레임을 나타낸다. 첫 번째 실험에서는 비교적 느린 속도로 물체가 이동하며, 물체가 영상을 벗어나지 않고 적은 움직임을 가진 구간이다.

본 실험에서는 물체의 움직임이 단순하기 때문에 두 알고리즘 모두 거의 모든 영역에서 정확한 추적이 이루어진 것을 확인할 수 있다. 하지만 기존의 CAMSHIFT 알고리즘의 경우, 190~220 프레임 사이에서 물체가 탐색 윈도우를 벗어나 추적에 실패하였다가 이후 다시 추적이 이루어진 것을 확인할 수 있다.

4.2 물체의 움직임이 큰 영상에서의 실험

그림 8은 250~500 프레임 사이의 결과를 나타낸다. 이 구간에서는 물체가 빠른 속도로 이동하며, 물체가 영상을 완전히 벗어나는 프레임이 발생된다. 프레임 구간에서 ‘-’ 부분은 물체가 영상 내에 존재하지 않는 프레임을 나타낸다.

330~340 프레임 영역에서는 물체가 영상 좌측 상단의 매우 어두운 지역에 존재하여 색상의 인식이 불가능한 지역

표 1. 테스트 영상에서의 대상 물체 검출 및 추적 성공률.

Table 1. Success rates of object detection and tracking in test image frames.

	기존의 CAMSHIFT	제안된 CAMSHIFT 기반
1~250 프레임 (물체의 움직임이 적고 속도가 느림)	84 %	100 %
250~500 프레임 (물체의 움직임이 크고 속도가 빠름)	38 %	85.7 %

으로 물체의 추적이 실패하였다. 기존의 CAMSHIFT 알고리즘은 340 프레임에서부터 물체의 탐색이 더 이상 이루어지지 않은 것을 확인 할 수 있다.

한편, 제안된 알고리즘의 경우 물체의 추적이 실패하면 지속적으로 영역 분할에 의한 초기 탐색 윈도우 설정방법을 시도하기 때문에 추적이 실패하거나 물체가 영상을 완전히 벗어난 후에도 빠르게 재탐색이 이루어 짐을 확인할 수 있다. 표 1에 테스트 영상에서의 물체 검출 및 추적 실험의 성공률 결과에 대한 비교 수치가 나와 있다.

4.3 다양한 물체에 대한 추적 실험

그림 9는 실생활에서 흔히 볼 수 있는 물체들로, 앞의 실험에서 사용한 파란색 공과는 형태와 색상이 다르다. 실험은 전체 1,000 프레임의 영상을 대상으로 수행하였다. 500 프레임 이전에는 상대적으로 물체의 움직임을 작게 하였으며, 500 프레임 이후에는 물체의 움직임을 크게 하였다.

그림 10은 다각형 형태의 분무기를 추종 대상으로 한 추적 실험 결과이다. 추적에 사용한 부분은 분무기 상단의 파란색 부분이다.

1~352 프레임 영역에서는 두 알고리즘 모두 분무기를 추적하였으나, 기존의 CAMSHIFT 알고리즘은 353 프레임부터 분무기를 추적하지 못하였다. 367 프레임부터 일시적으로 분무기를 추적하였으나, 이는 352 프레임에서 추적한 위치에 근접하였기 때문에 가능했던 것으로 확인하였다. 그러나 물체의 움직임이 커진 500 프레임 이후에는 기존의 CAMSHIFT 알고리즘의 경우 물체를 추적하지 못하였다. 제안된 CAMSHIFT 기반 알고리즘의 경우 일부 구간에서는 추적을 실패하였지만, 지속적으로 재탐색이 이루어진 것을 확인할 수 있다.

그림 11은 노란색 붓을 대상으로 한 추적 실험으로, 움직임에 따라서 형태가 달라지는 특징이 있다. 분무기의 실험 결과보다는 두 알고리즘의 추적 성공률이 높게 나타났

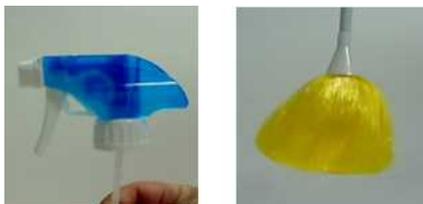


그림 9. 추적 대상(분무기, 붓).
Fig. 9. Tracking objects(Spray, Brush).

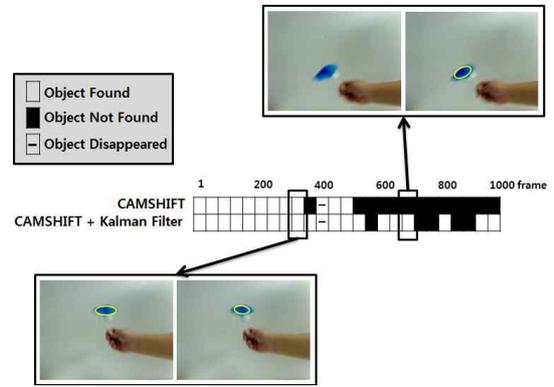


그림 10. 분무기 추적 실험 결과.
Fig. 10. Experiment results of spray tracking.

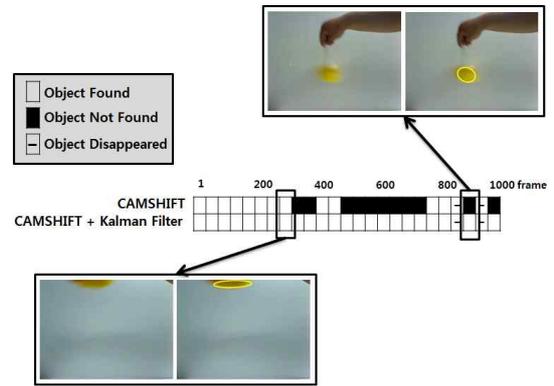


그림 11. 붓 추종 실험 결과.
Fig. 11. Experiment results of brush tracking.

표 2. 다양한 물체에 대한 검출 및 추적 성공률.
Table 2. Success rates of detection and tracking for various objects.

	기존의 CAMSHIFT	제안된 CAMSHIFT 기반
분무기	52 %	88 %
붓	71 %	93 %

다. 이는 붓의 크기가 크고, 형태의 변화가 크지 않았기 때문으로 분석된다.

그림 11에 나타난 것처럼 300 프레임에서는 붓의 일부만이 영상에 나타났다. 물체의 움직임이 작더라도, 기존의 CAMSHIFT는 붓을 검출하지 못하였지만, 개선된 CAMSHIFT는 붓을 검출하는데 성공하였다. 이는 칼만 필터에 의하여 물체의 이동 방향에 따라서 탐색 윈도우가 제어되었기 때문으로 분석된다. 노란색 붓에 대한 추적 실험은 분무기와 마찬가지로 물체의 움직임이 큰 영상에서 성능차이를 보였다. 표 2에 2가지 물체에 대한 검출 및 추적 실험 성공률이 정리되어 있다.

V. 결론

본 논문에서는 기존 물체 추적에 사용되던 CAMSHIFT 알고리즘의 문제점을 분석하고, 이를 보완하기 위해 이동 물체의 위치를 예측하기 위한 칼만 필터와 효율적인 추적

성능을 위한 가변적인 탐색 윈도우 설정 기법을 제안하였다. 또한 사용자에게 의해 설정해야 했던 초기 탐색 윈도우를 역투영 영상의 분할 영역에서 색상 분포를 통해 자동적으로 설정하도록 하며, 추적이 실패하였을 때 이를 통해 재탐색을 수행하여 지속적인 탐색이 가능하도록 하였다.

제안된 기법들에 대한 사전 실험을 통해 효율적이고 강한 검출 성능을 확인할 수 있었으며, 다양한 물체와 이동 속도의 변화등을 포함한 실험에서 기존의 CAMSHIFT 알고리즘보다 제안된 알고리즘의 검출 및 추적 성능이 크게 향상된 것을 확인할 수 있었다.

하지만 색상을 기반으로 알고리즘이 수행되기 때문에 빛이나 조명등에 의해 색상의 검출이 실패하였을 경우, 혹은 동일 색상의 분포가 많을 경우 추적이 어려운 문제점도 확인하였다.

향후 적용된 칼만 필터의 정확성을 높이고 색상 검출에서 발생하는 문제점을 해결하기 위한 컬러 모델이나, 대상 물체의 형태 정보를 적용하여 더욱 강한 추적 성능을 도모하고, 실제 보안 및 감시 시스템이나 로봇에의 적용이 필요하다고 생각된다.

참고문헌

- [1] N. D. Binh, "A robust framework for visual object tracking," *International Conf. on Computing and Communication Technologies*, pp. 1-8, 2009.
- [2] L. Rui, D. Zhijiang, H. Fujun, K. Minxiu, and S. Lining, "Tracking a moving object with mobile robot based on vision," *International Conf. on Neural Networks (IJCNN '08)*, pp. 716-720, 2008.
- [3] T. Jin and H. Tack, "A study on kohonen network based on path determination for efficient moving trajectory on mobile robot," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 10, no. 2, pp. 101-106, 2010.
- [4] D. Lee, H. Jeon, and Y. Joo, "Collaborative tracking algorithm for intelligent video surveillance systems using multiple network cameras," *Journal of Korean Institute of Intelligent Systems (in Korean)*, vol. 21, no. 6, pp. 743-748, 2011.
- [5] L. Davis, V. Philomin, and R. Duraiswami, "Tracking humans from a moving platform," *International Conf. on Pattern Recognition*, vol. 4, pp. 171-178, 2000.
- [6] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345-352, 2009.
- [7] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, pp. 2142-2151, 2000.
- [8] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, Q2, 1998.
- [9] P. Vadakkepat, P. Lim, L. C. De silva, L. jing, and L. L. Ling, "Multimodal approach to human-face detection and tracking," *IEEE Trans. on Industrial Electronics*, vol. 55, no. 3, pp. 1385-1393, 2008.
- [10] D. Exner, E. Bruns, D. Kurz, A. Grundhofer, and O. Bimber, "Fast and robust CAMShift tracking," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 9-16, 2010.
- [11] G. Bishop and G. Welch, "An introduction to the Kalman filter," *SIGGRAPH*, Course 8, 2001.
- [12] X. Gang, C. Yong, C. Jiu-jin, and G. Fei, "Automatic camshift tracking algorithm based on fuzzy inference background difference combining with twice seraching," *International Conf. on E-Health Networking, Digital Ecosystems and Technologies*, pp. 1-4, 2010.
- [13] C. Zhang, Y. Qiao, E. Fallon, and C. Xu, "An improved camshift algorithm for target tracking in video surveillance," *Conf. of 9th. Information Technology & Telecommunication*, pp. 19-26, 2009.
- [14] Y. Ling, J. Zhang, and J. Xing, "Video object tracking based on position prediction guide CAMSHIFT," *3rd International Conf. on Advanced Computer Theory and Engineering*, pp. 159-164, 2010.
- [15] Intel Open Source Computer Vision Library. <http://opencv.willow-garage.com/wiki/>



김 영 균

2008년 서경대학교 전자공학과 공학사. 2011년 서경대학교 대학원 전자공학과 공학석사. 2011년~현재 현대엠엔소프트 연구원. 관심분야는 GP, 영상인식, 지능로봇.



현 병 용

2011년 서경대학교 전자공학과 공학사. 2012년~현재 동 대학원 전자공학과 석사과정. 관심분야는 GA, GP, 영상인식, GPU 진화연산.



조 영 완

1991년 연세대학교 전자공학과 졸업. 1993년 연세대학교 대학원 전자공학과 공학석사. 1999년 동 대학원 공학박사. 2000년~2003년 삼성전자 디지털프린팅 사업부 책임연구원. 2003년~현재 서경대학교 컴퓨터공학과 조교수. 관심분야는 퍼지 및 신경망 시스템, 적응 및 강인 제어, 보행로봇의 자세 제어, 무인 이동체 시스템.



서기성

1986년 연세대학교 전기공학과 공학사.
1988년 연세대학교 전기공학과 공학석사.
1993년 연세대학교 전기공학과 공학박사.
1993년~1998년 서경대학교 산업공학과, 전자공학과 조교수.
1999년~2003년 Michigan State University,

Genetic Algorithms Research and Applications Group, Research Associate.
2002년~2003년 Michigan State University, Electrical & Computer Engineering, Visiting Assistant Professor.
2004년~현재 서경대학교 전자공학과 부교수. 관심분야는 진화 로봇틱스, GP기반 로봇보행, CPG기반 로봇보행, 물체인식과 추적, 진화최적화.