

Study on a Robust Object Tracking Algorithm Based on Improved SURF Method with CamShift

Hyochang Ahn*, In-Kyoung Shin**

Abstract

Recently, surveillance systems are widely used, and one of the key technologies in this surveillance system is to recognize and track objects. In order to track a moving object robustly and efficiently in a complex environment, it is necessary to extract the feature points in the interesting object and to track the object using the feature points. In this paper, we propose a method to track interesting objects in real time by eliminating unnecessary information from objects, generating feature point descriptors using only key feature points, and reducing computational complexity for object recognition. Experimental results show that the proposed method is faster and more robust than conventional methods, and can accurately track objects in various environments.

▶ Keyword: Object Tracking, Feature Matching, Object Detection, SURF, CamShift

1. Introduction

최근 여러 상황을 감시하고 신속하게 대응하기 위해 영상 감시 및 보안 모니터링 시스템 기술이 급격하게 발전하였다 [1,2,3]. 감시 시스템 기술 발전으로 중요시설 및 공공장소에서 치안 및 보안을 위해 감시 카메라를 설치하는 곳이 증가하고 있다. 이러한 감시 시스템에서 중요한 핵심 기술 중의 하나는 물체를 인식하고, 인식한 물체를 추적하는 기술이다[4]. 물체의 특징점 추출은 컴퓨터 비전 분야에서 많이 사용되는 기술 중에 하나이다. 복잡한 환경에서 움직이는 물체를 강인하면서도 효율적으로 추적하기 위해서는 관심 물체의 특징점을 추출하고, 이를 활용하여 물체를 추적하는 기술이 필요하다[5,6,7].

다양한 환경에서 강인한 물체 인식 및 추적은 크게 두 단계로 구성될 수 있다. 첫 번째는 관심 대상에 대한 특징점을 추출하고, 특징점 기술자를 생성하는 것이다[5,8,9]. 두 번째는 생성된 특징점 기술자를 사용하여 입력 영상에서 대상 물체를 인식하고 지속적으로 물체를 추적하는 것이다[3,5,10]. 특징점을 기반으로 한 물체 추적은 때때로 물체가 급작스럽게 이동하는 경우에도 특징점 정합을 통해서 물체를 인식하고 추적하는 것이 가능하다. 그러나 특징점에 기반을 둔 시스템에서는 특징

점 추출이 전체 시스템의 성능을 크게 좌우하게 된다 [4,11,12]. 대상 물체에서 많은 양의 특징점이 추출되면 물체를 정확하게 인식할 수 있다. 그러나 특징점 추출 및 정합 과정에서 연산량이 크게 증가하게 되어, 복잡한 환경에서 물체를 실시간 인식하고 추적하는 것이 불가능하다[8,13]. 반면, 추출되는 특징점의 양이 적으면, 정확한 특징점 기술자를 생성하기 어렵게 되어 물체를 정확하게 인식하기 힘든 문제점이 발생하게 된다. 대부분의 특징점 정합(Feature Matching)에 기반을 둔 물체 추적 기법은 특징점 추출과 정합에 있어 많은 연산량이 필요하기 때문에 실시간으로 물체를 추적하기에 부적절하다 [12,13]. 따라서 대상 물체에서 불필요한 정보를 제거하여 핵심적인 특징점만을 활용하여 특징점 기술자를 생성하면, 물체 인식에 대한 계산 복잡도를 줄일 수 있어, 실시간으로 관심 물체를 추적하는 것이 가능하다.

관심 물체를 추적하는 기법은 데이터 집합의 특징점 정보, 코너 정보 및 색상 정보 등의 밀도 분포를 기반으로 하여 관심영역의 물체를 고속으로 추적하는 MeanShift 방법이 주로 사용된다 [11,14,15]. 최초로 물체를 인식하면 반복되는 색상 분할 계산에

* First Author: Hyochang Ahn, Corresponding Author: In-Kyoung Shin

*Hyochang Ahn (youcu92@gmail.com), Dept. of Smart & PhotoVoltaic Convergence, Far East University

**In-Kyoung Shin (shingguri@gmail.com), Dept. of Applied Computer Engineering, Dankook University

• Received: 2017. 12. 15, Revised: 2018. 01. 05, Accepted: 2018. 01. 23.

의해서 색상 클러스터를 발생하고, 선택된 물체의 색상 영역에서 경계를 결정하여 관심 물체를 추적한다[16]. 그러나 MeanShift 방법은 화소값 히스토그램만으로 물체를 추적하기 때문에 물체와 유사한 색상의 배경이 있을 경우 추적이 잘 되지 않는 문제점과 물체의 크기 변화나 탐색 윈도우의 크기가 부적절할 경우 추적이 중단되는 문제점을 가지고 있다. 반면, CamShift(Continuously Adaptive MeanShift) 방법은 MeanShift 방법의 단점을 보완하기 위해 탐색 윈도우가 윈도우 내의 화소값들의 분류에 따라 계속 변화시킨다. 탐색이 종료되면 최종 윈도우를 객체의 영역으로 결정하여 물체를 추적함으로써 MeanShift 방법의 단점을 보완한다 [4,11,17,18]. 본 논문에서는 복잡한 환경에 강인하고 향상된 특징점 추출 방법을 활용하여 실시간 환경에서 물체 인식하고 추적하는 기법을 제안한다.

본 논문의 나머지 부분의 구성은 다음과 같다. 2장에서는 물체 인식을 위한 특징점 추출 및 추적에 대한 관련 연구를 간략하게 소개한다. 3장에서는 제안한 방법에 대해 설명한다. 4장에서는 제안한 방법에 대한 실험 및 결과를 보여준다. 마지막으로 5장에서는 결론과 향후 연구를 제시한다.

II. Related works

영상 감시 시스템은 일반적으로 물체 인식과 추적으로 구성된다. 물체 인식에서 가장 중요한 것은 물체를 정확하게 추출하는 것이다. 특징점 기술자는 물체를 정확하게 추출하기 위해서 조도, 회전, 크기 변화, 어파인 변화(Affine Transformation)에 강인해야 한다 [13,14,15]. 강인한 특징점 기술자를 생성하기 위해 사용되는 특징점 추출 방법은 일반적으로 SIFT (Scale Invariant Feature Transform) 방법이 있다[[19,20,22]. SIFT 방법은 저수준의 특징점 추출과 이를 이용한 이미지 정합 문제를 해결하는 것을 목적으로 물체의 크기, 회전 및 투영에 대한 강건한 특성을 가진 지역적 특징의 집합으로 변환하는 방법이다[23,24].

SIFT 방법은 스케일 공간 극값 검출, 키포인트 지역화, 방향 할당 및 키포인트 기술자 생성의 4단계로 구성된다. 먼저 스케일 공간 극값 검출은 스케일과 시점을 변화시키지 않고 분산 스케일을 조절하여 영상에서 강인한 특징점 후보를 검출하는 것이다. SIFT는 가우시안 함수를 사용하는 대신 DoG (Difference of Gaussian) 함수를 사용하여 계산 속도를 향상시키면서 강인한 특징점을 검출한다. DoG 피라미드는 수식 (1)과 같이 가우시안 커널과 영상을 회전 연산(Convolution)을 수행하여 생성할 수 있다.

$$L(x,y,\sigma) = G(x,y,\sigma) \otimes I(x,y) \quad (1)$$

현재 위치의 8개의 이웃한 화소와 상하로 인접한 3×3 공간에서 화소쌍을 이용하여 총 26개의 화소와 비교하여 특징점 후보로

결정한다. 키포인트 지역화 단계에서는 정확한 정합을 수행하기 위해 보정하는 단계로, 낮은 대비를 갖는 특징점 후보를 제거한다. SIFT는 모서리에서 생성되는 특징점을 중요시하여 윤곽선에서 생성되는 특징점 후보를 제거한다. 이를 위해서 헤이시안 행렬(Hessian Matrix)을 사용하여 특징점 후보에 대한 고유값을 계산하고 임계값을 사용하여 특징점 후보에서 제외시킨다. 방향 할당 단계에서는 특징점의 방향과 크기를 구하는 단계이다. 이전 단계에서 선택된 특징점을 기준으로 이웃 화소의 크기와 방향을 가진 벡터를 생성한다. 이를 통해 방향 히스토그램을 구성하고, 빈도수가 가장 높은 곳을 찾아 특징점의 주 방향으로 결정한다. 마지막 단계인 키포인트 기술자 생성에서는 결정된 주방향에 대해서 4×4 영역으로 나누어 방향 히스토그램을 재구성한다. 이렇게 생성된 방향 히스토그램은 각각 8개의 방향과 크기를 가지는 키포인트 기술자로 생성된다.

SIFT 방법은 고차원 특성 때문에 이미지의 크기가 커짐에 따라 데이터양이 크게 증가하게 된다. 데이터양이 증가함에 따라 연산량 또한 증가하게 되며, 이에 따라 연산 시간이 증가하는 문제가 발생하게 된다. SURF 방법은 SIFT 방법에서 특징 추출과 정합 단계에서 근사화 방법을 사용하여 연산의 복잡성을 줄일 수 있어, 빠르게 특징점을 추출하고 특징점 기술자를 생성할 수 있다[25,26]. 또한 SURF 방법은 SIFT 방법 보다는 물체 인식에 있어 연산량을 줄일 수 있음에도 불구하고, 실시간으로 물체 인식하기에는 연산량이 많은 단점을 가지고 있다. 따라서 복잡한 환경에 강인하면서도 정확하게 물체를 인식할 수 있는 특징점 추출 및 기술자를 생성할 수 있는 방법이 필요하다.

물체를 추적 기술은 주로 MeanShift 방법이 많이 사용된다 [16,27]. MeanShift 방법은 특징점, 코너, 색상 등의 데이터 집합을 기반으로 하여 특정 데이터들의 중심을 찾고 이동시켜 관심 영역에서 물체를 인식하고 고속으로 추적하는 방법이다. 초기에 관심 물체에 대한 영역을 설정하고, 설정된 영역에서 반복되는 색 분할을 계산하여 색상 클러스터를 생성한다. 생성된 색상 클러스터에 기반을 두어 경계를 결정하면, 관심 물체를 추출할 수 있다. MeanShift 방법을 통해 물체를 추적하는 방법은 다음과 같다. MeanShift 방법은 관심영역을 설정하고 선택된 범위 안에 있는 특징점의 밀도를 분석하여 가장 큰 밀도를 찾고 밀도가 가장 큰 지점을 중심으로 재설정 한다. 그리고 재설정된 중심점을 기준으로 관심 범위를 다시 설정한다. 이러한 방법을 반복하여 전체에서 가장 밀도가 높은 윈도우를 찾아 이동하면서 물체를 추적한다. 그러나 물체의 색상과 유사한 경우나 물체의 크기 변화 및 부적절한 탐색 윈도우의 크기로 인하여 가장 밀도가 높은 윈도우에서 맴돌 수 있는 문제점을 가지고 있다. 반면 CamShift 방법은 MeanShift 방법을 변형한 것으로, 특정 물체를 추출하기 위해 경험적 분포를 통하여 동작하는 탐색 방법이다 [28,29,30,31]. 초기에 탐색 윈도우의 크기를 정하고, 탐색 윈도우를 초기 중심점의 중앙에 오도록 위치시킨다. 다음 중심점을 찾기 위해 0차 모멘트(Moment)를 구하고, x 와 y 에 대한 1차 모멘트를 각각 구한다. 이러한 방법으로 새로운 중심점을 구하고,

새로운 탐색 윈도우의 중심점의 좌표를 구한다. 새로운 탐색 윈도우에 대하여 위의 과정을 MeanShift 방법이 수렴할 때까지 반복하여 물체를 추적한다. 따라서 CamShift는 물체를 고속으로 추적하는데 사용되지만, 조도변화나 복잡한 환경에서는 추적 성능이 좋지 않은 단점을 가지고 있다.

III. Proposed Method

CamShift 방법은 실시간 환경에 대한 관심 물체를 추적하는 일반적인 방법이다. 그러나 CamShift 방법은 색상 특징만 사용하여 주위 환경과 조명에 강건하지 않다. 따라서 CamShift 방법은 조명과 주위 잡음에 민감한 특징을 가지게 되어, 관심 물체가 배경과 유사한 색상을 가지고 있을 때 관심 대상을 잃어버리는 문제점을 가지게 된다. 이러한 문제점은 SURF 방법을 사용하여 특징점을 추출하고 특징점 기술자를 사용하여 대상 물체를 인식하고 CamShift 방법으로 물체를 추적 할 수 있는 통합 된 방법으로 문제를 해결할 수 있다. SURF 방법은 회전과 크기 변화에 불변한 특징점을 사용하여 물체를 찾을 수 있는 장점을 가지고 있기 때문에 카메라의 각도와 거리에 관계없이 동일한 대상을 인식 할 수 있는 특징을 가지고 있다. 실시간 환경에서는 빠르면서도 정확하게 물체를 인식하기 위해 계산의 복잡도를 줄이는 방법으로 효율인 정합점을 찾는 것이 필요하다. 그러나 기존의 SURF 방법은 SIFT 방법보다는 계산 복잡도가 낮지만, 실시간으로 물체를 인식하는 경우에는 적합하지 않다.

따라서 본 논문에서는 관심 대상의 특징점을 추출하고 불필요한 특징점을 제거하여 특징점 기술자의 차원을 효율적으로 줄임으로써 연산의 복잡도를 감소시킴으로써, 빠르면서도 효율적으로 물체 인식 및 추적하는 방법을 제안한다. 제안한 특징점 기반의 물체 추적 시스템은 다음과 같은 과정으로 진행된다. 관심 물체의 특징점을 추출하고 필요한 특징점만을 선별하여 특징점 기술자를 생성한다. 생성된 기술자를 사용하여 영상에서 관심 물체를 인식하고, 인식된 부분에 대해 추적을 위한 탐색 윈도우를 생성한다. 생성된 탐색 윈도우의 중심을 찾고 밀도를 분석하여 밀도가 가장 큰 지점을 중심점으로 재설정 한다. 재설정된 중심을 기준으로 관심 범위를 다시 설정하고 위의 방법을 반복하여 전체에서 가장 밀도가 높은 윈도우를 찾아 이동하면서 물체를 추적한다. 이때 물체를 잃어버리면 기술된 특징점을 활용하여 다시 영상에서 관심 물체를 찾아 일련의 과정을 반복한다.

1. Object Recognition

특징점 기반의 물체 인식에서 입력 이미지의 크기 및 회전과 같은 왜곡에 강인한 특징점을 추출하는 것이 중요한 요소중에 하나이다. 기존의 특징점 추출 방법들은 특징점 추출 및 인식면에 높은 성능을 보여주고 있다. 그러나 특징점 추출 및 기술자 생성에 대한 연산 복잡도가 높기 때문에 수행시간이 오래 걸리

는 단점을 가지고 있다. 따라서 본 논문에서는 SURF 방법 기반의 적응적 특징점 추출 및 기술자를 생성하여 이미지의 기하학적 왜곡에 강인하면서도 연산량이 적은 특징점 추출 방법을 제안한다. SURF 방법은 근사화된 헤이시안 행렬을 적용한 고속 헤이시안 검출기를 사용하여 특징점을 고속으로 검출하고 적분 이미지를 사용하여 복잡한 계산량을 크게 감소시켰다.

SURF 방법은 박스 필터를 사용하여 반복적인 회전 연산을 수행하여 특징점을 고속으로 추출한다. 이때 스케일 공간상의 연산량을 감소시키기 위해 적분 이미지를 사용한다. 적분 이미지를 생성하여 연산을 수행하면, 임의의 크기의 이미지에서 관심 영역을 선택하더라도, 관심 영역 내의 모든 화소값의 합을 간단한 연산을 통해 빠르고 정확하게 계산할 수 있는 특징을 가지게 된다. 적분 이미지는 수식 (2)을 통해서 생성할 수 있다. 입력 이미지의 사각형의 관심 영역 내에서 모든 화소값을 가산함으로써 적분 이미지를 생성한다.

$$I_{int}(X) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (2)$$

수식 (1)에서, $X=(x,y)^T$ 의 위치에서 적분 이미지는 $I_{int}(X)$ 과 같이 표현된다. 입력 이미지의 화소값은 $I(i,j)$ 으로 표시한다. 적분 이미지는 계산 영역에 대한 종속성이 없어 크기가 큰 필터 사이즈를 반복적으로 사용하면 효율적인 연산을 수행할 수 있다. 관심 영역 내의 모든 화소값의 합은 수식 (2)과 수식 (3)를 통하여 계산된다. 수식 (3)는 관심영역 D 의 적분 값 계산 과정을 나타내고 있다. 이때, A 의 영역은 B 와 C 를 공통으로 포함하는 영역이다.

$$I_D = I_K - (I_B + I_C - I_A) \quad (3)$$

특징점 추출은 먼저 이미지의 특징으로 사용될 가능성이 높은 후보군을 추출하는 과정을 수행한다. 후보군 추출은 기존의 특징점 추출 방법인 SIFT 방법에서는 가우시안 필터를 사용하여 원본 이미지를 사용하여 회전 연산을 수행한다. 따라서 레벨이 올라갈 수록 이미지의 크기를 1/4로 줄여 가면서 회전 연산을 수행한다. 반면에, SURF 방법은 특징점을 고속으로 추출하기 위해서 가우시안 필터를 사용하는 대신 박스 필터를 활용한 헤이시안 행렬 기반의 특징 추출기를 사용한다. 따라서 회전 연산을 수행할 때, 원본 이미지의 크기를 변화시키지 않고, 박스 필터의 크기를 다양하게 증가시켜 사용하기 때문에 고속으로 특징점을 추출할 수 있다. 수식 (4)과 수식 (4)는 입력 이미지와 2차 미분된 가우시안 필터를 회전 연산을 통해 헤이시안 행렬을 구하는 것을 나타내고 있다.

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \quad (4)$$

$$L_{xx}(X, \sigma) = I(x, y) * \frac{\partial^2}{\partial x^2} g(\sigma) \quad (5)$$

여기서 $L_{xx}(X, \sigma)$ 는 입력 이미지의 $X=I(x, y)$ 인 화소에 σ 의 스케일을 갖는 수평 방향인 xx 에서 2차 미분된 가우시안 필터 및 입력 연산의 회선 연산을 나타낸다. 또한, $L_{xy}(X, \sigma)$ 와 $L_{yy}(X, \sigma)$ 는 가우시안 필터를 대각선 방향을 나타내는 xy 와 수직 방향을 나타내는 yy 방향에 대해 2차 미분한 값과 이미지에 대해 회선 연산 나타낸다. 그러나 가우시안 필터는 연산 복잡도가 높아 필터를 적용하는 시간이 많이 걸린다는 문제점을 가지고 있다. 이러한 문제점을 SURF 방법에서는 수식 (6)와 같이 가우시안 2차 미분의 회선 연산을 근사화한 박스 필터를 적용하여 해결한다.

$$\det(H_{app}) = D_{xx}D_{yy} - (\omega D_{xy})^2 \quad (6)$$

여기서 연관 상수 ω 는 헤이시안 행렬의 균형을 맞추기 위해 필요하다. 이 연관 상수는 수식 (7)과 같이 프로베니우스 놈(Frobenius Norm)을 사용한다. 이와 같은 방법으로 연산량을 크게 감소시켜, 필터 크기가 같은 경우 처리 속도면에서 보다 좋은 성능을 보인다.

$$\omega = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912 \cong 0.9 \quad (7)$$

전통적인 SURF 방법으로 특징점을 추출하는 경우 불필요한 특징점도 추출되어 관심 물체에 대한 많은 특징점이 생성된다. 이미지에서 불필요한 정보가 많은 경우, 추출되는 특징점이 많게 되고, 특징점 기술자를 생성하고 물체를 인식하는 과정에서도 연산량에 대한 복잡도가 높아지게 되어 정합 속도가 떨어지게 된다. 반대로, 특징점의 수가 너무 적게 되면, 특징점 기술자를 생성하고 물체를 인식하는 과정에서 정합 정보가 부족하여 정합을 물체 인식을 정확하게 수행할 수 없게 된다. 따라서 본 논문에서는 추출된 특징점에서 필요한 정보만을 추출하고, 특징점 기술자를 생성하여, 효율적인 물체 인식을 할 수 있는 적응적 특징점 추출 방법을 제안한다. 제안한 방법은 추출된 특징점 중에서 일정 범위 내에 있지 않은 특징점은 잡음 등으로 간주하여, 탐색 윈도우 안에 적은 양의 특징점이 있는 경우 불필요한 정보로 간주하여, 특징점 기술자 생성시 이를 제외시켜 수행 속도를 향상시킬 수 있다. 수식 (8)과 수식 (9)은 적응적 특징점을 추출하는 방법을 나타내고 있다.

$$s^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \quad (8)$$

$$G = \begin{cases} 1, & N > T_N \text{ and } s < T_s \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

여기서 σ 는 적응적 특징점 추출을 위한 탐색 윈도우 안의 특징점 밀도를 나타낸다. N 은 각 탐색 윈도우 안에 있는 특징점의 개수를 나타내고, X_i 는 특징점의 좌표(x_i, y_i)를 나타낸다. 또한 \bar{X} 는 특징점 탐색 윈도우의 중앙을 표시하고, T_N 은 탐색 윈도우의 크기에 대한 임계치를 의미하며, T_s 는 탐색 윈도우 안에 추출된 특징점의 밀집도 임계치를 나타낸다. G 는 적응적 특징점 추출을 통한 특징점 사용 여부를 표시한다.

2. Object Tracking

CamShift 방법은 색상 기반으로 물체를 추적하는 MeanShift 방법에서 변형되었다. CamShift 방법은 MeanShift 방법의 다음 프레임의 초기 값으로 마지막 프레임의 결과를 만들고 반복하는 단계를 수행한다. CamShift 방법의 실행 과정은 다음과 같다.

관심 대상 물체의 크기가 변경되면 CamShift 방법은 물체 추적을 계속하기 위해 물체 탐색 영역을 물체의 크기에 따라 적응적으로 크기를 조절한다. 이미지 전체에서 확률 분포를 계산할 영역을 설정하고, MeanShift 방법의 탐색 윈도우의 초기 위치를 설정한다. 초기 위치를 설정한 후 0차 모멘트와 1차 모멘트를 계산한다. 수식 (10)를 사용하여 관심 영역에 대한 모멘트를 계산한다. 수식에서 M_{00} 는 0차 모멘트를 나타내고, M_{10} 와 M_{01} 는 x 와 y 에 대한 1차 모멘트를 나타낸다.

$$\begin{aligned} M_{00} &= \sum_x \sum_y I(x, y) \\ M_{10} &= \sum_x \sum_y x I(x, y) \\ M_{01} &= \sum_x \sum_y y I(x, y) \end{aligned} \quad (10)$$

수식 (10)에서 $I(x, y)$ 는 탐색 윈도우 내에 위치한 (x, y) 의 역투영 확률 분포를 나타낸다. x 와 y 는 탐색 윈도우의 범위를 나타내고, MeanShift 방법의 탐색 윈도우 보다 약간 크다. 0차 모멘트와 1차 모멘트를 계산한 후, 관심 대상 물체의 중심에 대한 평균 위치를 계산한다. 평균 중심 위치는 수식 (11)으로 표현할 수 있다.

$$x_c = \frac{M_{10}}{M_{00}}, \quad y_c = \frac{M_{01}}{M_{00}} \quad (11)$$

x_c 와 y_c 는 x 와 y 의 중심 값을 나타낸다. 다음 비디오 프레임에서 탐색 윈도우의 크기는 0번째 모멘트의 컬러 확률 분포 함수를 사용하여 계산한다. 마지막으로, 중심점의 변화가 사전에 정의된 임계값보다 작은 경우, 수렴시키기 위해서 위의 과정을 반복 수행한다. 물체를 추적한 후, 탐색 윈도우의 밝기 값에

대한 0번째 모멘트와 1번째 모멘트를 계산하여 관심 영역에서 물체의 크기와 각도를 계산한다. x 와 y 의 두 번째 모멘트는 수식 (12)을 통해 계산할 수 있다.

$$\begin{aligned} M_{20} &= \sum_x \sum_y x^2 I(x,y) \\ M_{02} &= \sum_x \sum_y y^2 I(x,y) \\ M_{11} &= \sum_x \sum_y xy I(x,y) \end{aligned} \quad (12)$$

탐색 윈도우를 다시 계산해야 하는 경우, 밝기 값에 대한 분포를 계산하여 수직, 수평 축의 크기를 갱신하는 것이 필요하다.

IV. Experiments and Results

본 논문에서 제안한 방법의 성능 평가를 위해서 사용된 환경은 Windows 10에서 Visual Studio 2017 사용하여 실험하였다. 또한 NTU-VOI 데이터셋을 OpenCV 라이브러리를 활용하여 관심 물체를 검출하고 추적 알고리즘을 구현하였다[32,33]. 그림 1은 실험환경에서 제안한 방법이 구동되고 있는 모습을 보여주고 있다. 실험은 관심 대상 물체의 특징점 추출 및 입력 영상과 정합하는 것에 중점을 두었다. 인식된 물체는 사각형으로 표시하였으며, 물체의 크기와 위치에 따라 인식된 사각형의 모양과 크기가 다양하게 나타나도록 하였다.

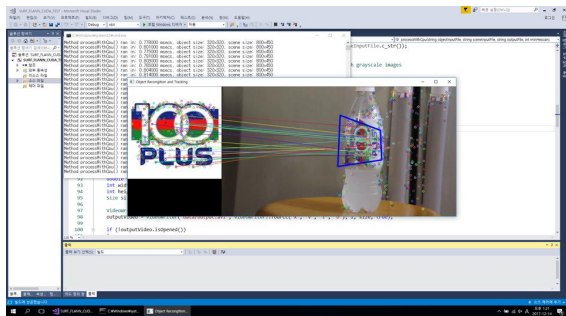


Fig. 1. Implementation Environment for Proposed Method

본 논문에서 수행한 관심 물체 인식 및 추적 과정은 아래에 나온 그림 2와 같다. 먼저 관심 물체의 특징점을 추출하고 기술자를 생성하여 탐색 과정을 거쳐 영상에서 물체를 인식하고 추적을 위한 탐색 윈도우를 생성한다. 생성된 탐색 윈도우의 중심을 찾고 밀도가 가장 큰 지점을 물체 추적을 위한 중심점으로 설정한다. 그리고 설정된 중심점을 기준으로 탐색 범위를 다시 설정한다. 이와 같은 방법을 반복하여 물체를 추적한다. 물체가 빠르게 이동하여 탐색 범위를 잃어버릴 경우, 생성된 특징점 기술자를 통해 다시 영상에서 물체를 인식하고 위의 과정을 반복하여 추적한다.

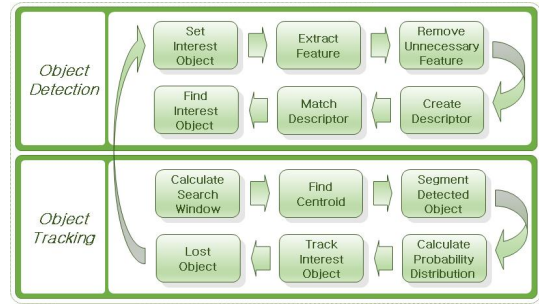


Fig. 2. Diagram for Proposed Method

그림 3에서 그림 5까지는 다양한 환경에서 제안한 방법으로 관심 대상을 입력 영상에서 인식하고 추적하는 결과를 보여 주고 있다. 그림 3에서 알 수 있듯이 관심 물체(100Plus Logo)를 심플한 배경 및 복잡한 배경에서도 잘 인식하고 추적하는 것을 알 수 있다. 또한 그림 3의 (b)에서는 관심 물체가 회전되어 있음에도 인식하고 추적하고 있는 결과를 보여 주고 있다. 그림 4의 (a)에서는 관심 물체의 회전 및 스케일 변화가 있어도 정확하게 물체를 인식하고 추적하는 것을 보여 주고 있다. 또한 물체가 가려져 일부만 노출되어도 정상적으로 인식하고 추적을 수행하는 것을 보여주고 있다. 그림 5에서도 관심 물체의 회전 및 스케일 변화에도 정확하게 물체를 추적하는 것을 알 수 있다. 특히 (b)에서는 관심 물체의 색상변화가 생겼지만 정상적으로 물체를 인식하는 것을 보여주고 있다.

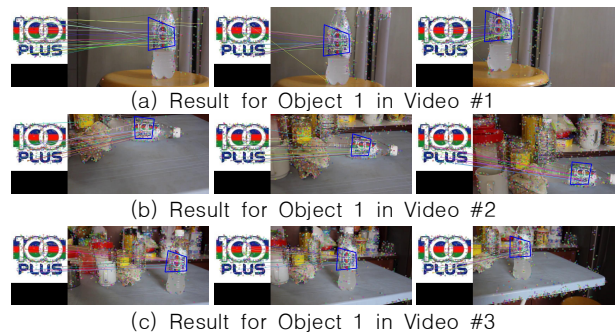


Fig. 3. The results of Object Detection and Tracking for Object 1(100Plus Logo)

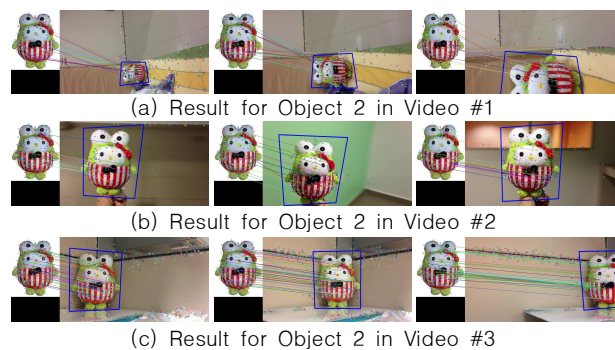


Fig. 4. The results of Object Detection and Tracking for Object 2(Kitty)



Fig. 5. The results of Object Detection and Tracking for Object 3(Starbucks Logo)

따라서 본 논문에서 제안한 방법이 동일한 대상을 다양한 조건하에서 정확하게 탐지하고 추적하고 있음을 보여주고 있다. 또한 물체의 회전, 이동, 크기 변화 및 가려짐 등의 환경과 배경이 복잡한 환경에서도 관심 대상을 정확하게 인식하고 추적하는 것을 알 수 있다.

Table 1. Experimental Result for Recognition Time(sec)

	Object 1	Object 2	Object 3
SIFT	3.92	3.12	3.21
SURF	1.29	1.20	1.48
Proposed	1.12	0.95	1.25

표 1은 제안한 알고리즘과 기존의 SIFT 방법 및 SURF 방법과의 관심 물체 인식 속도 차이를 보여 주고 있다. 기존 알고리즘인 SIFT 방법과는 많은 차이를 보여주고 있으며, SURF 방법과는 약 16% 정도 향상된 결과를 나타내고 있다. 따라서 제안한 방법은 불필요한 특징점을 제거하여 정합을 수행하므로 물체를 인식할 때 계산상의 복잡성이 높아지는 문제를 개선할 수 있었다. 또한 다양한 조건에서 대상 물체를 정확하게 인식하고 추적하여 배경에 비슷한 색이 있을 때 관심 대상을 잃어버려도 물체를 다시 추적할 수 있었다.

V. Conclusions

최근 여러 상황을 감시하고 신속하게 대응하기 위해 영상 감시 및 보안 모니터링 시스템 기술이 급격하게 발전하고 있으며, 관련 연구가 활발하게 진행되고 있다. 본 논문에서는 영상 감시 시스템에서 영상 내에서 움직이는 물체를 효율적으로 추적하기 위해 적응적 특징점 추출 및 정합을 사용하여 강인한 물체 인식 및 추적 방법을 제안하였다. 제안한 알고리즘은 불변의 특징을 갖는 객체를 인식하고, 불필요한 특징점을 제거하여 특징점 기술자의 차원을 감소하여 계산량을 감소시켰다. 실험 결과를 보면

제안한 방법이 기존의 방법보다 빠르고 강력하며 다양한 환경에서 정확하게 개체를 추적 할 수 있음을 보여주고 있다. 향후 연구로 제안한 방법을 활용하여 다중의 물체를 빠르게 탐지하는 다중 물체 기반의 감시 시스템에 대한 연구가 필요하다.

REFERENCES

- [1] J. Pons, J. Prades-Nebot, A. Albiol and J. Molina, "Fast motion detection in compressed domain for video surveillance.", *Electronics letters* Vol. 38, No. 9, pp. 409-411, 2002.
- [2] M. Tiwari and R. Singhai, "Improved Algorithm for Object Tracking in Video Camera Network", *Indian Journal of Science and Technology*, Vol. 10, No. 10, pp. 1-10, 2017.
- [3] M. M. Bhajibhakare and P. K. Deshmukh, "Detection and tracking of moving object for surveillance system.", *International Journal of Application of Innovation in Engineering and Management* Vol. 2, No. 12, pp. 298-301, 2013.
- [4] Q. Wang and Z. Gao, "Study on a Real-time Image Object Tracking System", *International Symposium on Computer Science and Computational Technology*, Vol. 2, pp. 788-791, 2008.
- [5] Z. Zhou, D. We, X. Peng, Z. Zhu and K. Luo, "Object Tracking Based on Camshift with Multi-feature Fusion", *Journal of Software*, Vol. 9, No. 1, pp. 147-153, 2014.
- [6] A. Yilmaz, O. Javed and M. Shah "Object Tracking: A survey", *ACM Computing Surveys*, Vol. 38, No. 4, Article. 13, 2006.
- [7] K. Werner, M. Kampel, "Interest point based tracking", *International Conference on Pattern Recognition*, pp. 3549-3552, 2010.
- [8] T. M. Barroso and P. F. Whelan, "Enhancing SURF Feature Matching Using Colour Histograms", *Irish Machine Vision and Image Processing Conf.*, pp. 111-112. 2011.
- [9] M. Du, J. Wang, J. Li, H. Cao, G. Cui, J. Lv and X. Chen, "Robot Robust Object Recognition based on Fast SURF Feature Matching", *Chinese Automation Congress*, pp. 581-586, 2013.
- [10] Y. Kim, W. Han, Y. H. Lee, C. G. Kim, and K. J. Kim, "Object Tracking and Recognition Based on Reliability Assessment of Learning in Mobile Environments". *Wireless Personal Communications*, Vol. 94, No. 2, pp. 267-282, 2017.
- [11] K. Du, Y. Ju, Y. Jin, G. Li, Y. Li and S. Qian, "Object Tracking based on Improved MeanShift and SIFT", 2nd

- Int. Conf. on Consumer Electronics, pp. 2716-2719, 2012.
- [12] J. Y. Zhang, H. Y. Wu, S. Chen and D. S. Xia, "The Target Tracking Method Based on Camshift Algorithm Combined with SIFT", *Advanced Materials Research*, Vol. 186, pp. 281-286, 2011.
- [13] H. Ahn and Y. Lee, "Performance analysis of object recognition and tracking for the use of surveillance system", *Journal of Ambient Intelligence and Humanized Computing* Vol. 7, No. 5, pp. 673-679, 2016.
- [14] S. A. Dave, M. S. Nagmode and A. Jahagirdar, "Statistical Survey on Object Detection and Tracking Methodologies", *International Journal of Scientific and Engineering Research*, Vol. 4, Issue. 3, 2013.
- [15] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, pp. 564-577, 2003.
- [16] I. Leichter, M. Lindenbaum and E. Rivlin, "Mean Shift tracking with multiple reference color histograms", *Computer Vision and Image Understanding*, Vol. 114, No. 3, pp. 400-408, 2010.
- [17] J. Wang, F. He, X. Zhang and Y. Gao, "Tracking Objects through Occlusions Using Improved Kalman Filter", *International Conference on Advanced Computer Control*, pp. 23-228, 2010
- [18] Y. Yue, Y. Gao and X. Zhang, "An Improved Camshift Algorithm Based on Dynamic Background", *International Conference on Information Science and Engineering*, pp. 1141-1144, 2009.
- [19] L. Juan and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF", *International Journal of Image Processing*, Vol. 3, Issue. 4, pp. 143-152, 2009.
- [20] S. Ha and Y. Moon, "Multiple Object Tracking Using SIFT Features and Location Matching", *International Journal of Smart Home*, Vol. 5, No. 4, pp. 17-26, 2011.
- [21] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 506-513, 2004.
- [22] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, Vol. 60, Issue. 2, pp. 91-110, 2004.
- [23] M. Brown and D. Lowe, "Invariant Features from Interest Point Groups", *British Machine Vision Conference*, pp. 656-665, 2002.
- [24] M. Stommel and O. Herzog, "Binarising SIFT-Descriptors to Reduce the Curse of Dimensionality in Histogram-Based Object Recognition", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, pp. 320-327, 2009.
- [25] H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding*, Vol. 110, Issue. 3, pp. 346-359, 2008.
- [26] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-518, 2001.
- [27] D. Exner, E. Bruns, D. Kurz and A. Grundhofer, "Fast and Robust CAMShift Tracking", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9-16, 2010.
- [28] N. Q. Nguyen, S. F. Su, Q. V. Tran, V. T. Nguyen, and J. T. Jeng, "Real time human tracking using improved CAM-shift" In *Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, 2017 Joint 17th World Congress of International, IEEE. pp. 1-5, 2017.
- [29] R. Stolkin, I. Florescu and G. Kamberov, "An Adaptive Background Model for Camshift with a Moving Camera", *6th Int. Conf. on Advances in Pattern Recognition*, pp. 261-265, 2007.
- [30] A. Basit, M. N. Dailey, P. Laksanacharoen, and J. Moonrinta, "Fast Target Redetection for CAMSHIFT using Back-projection and Histogram Matching", *Int. Conf. on Computer Vision Theory and Applications*, Vol. 3, pp. 507-514, 2014.
- [31] X. Liu, H. Chu and P. Li, "Research of the Improved Camshift Tracking Algorithm", *Int. Conf. on Mechatronics and Automation*, pp. 968-972, 2007.
- [32] A. R. J. Francois, "CAMSHIFT Tracker Design Experiments with Intel OpenCV and SAI", *International Conference on Pattern Recognition*, No.4, pp. 1-4, 2004.
- [33] <https://sites.google.com/site/jingjingmengsite/>

Authors



Hyochang Ahn received the M.S. and Ph.D. degrees in Electronics and Computer Engineering from Dankook University, South Korea, in 2006 and 2012, respectively. Dr. Ahn worked as Research Professor at Dankook University in the Department of

Applied Computer Engineering from 2014 to 2016. He is currently a Visiting Professor in Department of Smart & PhotoVoltaic Convergence, Far East University. He is interested in Image Processing, Computer Vision, Embedded system and Mobile Programming.



In-Kyoung Shin received the Ph.D. in Electronics and Computer Engineering from Dankook University, South Korea in 2014. Dr. Shin worked as served as a researcher at the App Center at Far East University from 2013 to 2015. She is

interests include Image Processing, Multimedia Application and Mobile Programming.