

## 증강현실 서비스를 위한 Camshift와 SURF를 개선한 객체 검출 및 추적 구현

이용환\*·김흥준\*\*†

\*원광대학교 디지털콘텐츠공학과, \*\*†경남과학기술대학교 컴퓨터공학과

### Implementation of Improved Object Detection and Tracking based on Camshift and SURF for Augmented Reality Service

Yong-Hwan Lee\* and Heung-Jun Kim\*\*†

\*Department Of Digital Contents, Wonkwang University

\*\*†Department of Computer Science and Engineering,  
Gyeongnam National University of Science and Technology

#### ABSTRACT

Object detection and tracking have become one of the most active research areas in the past few years, and play an important role in computer vision applications over our daily life. Many tracking techniques are proposed, and Camshift is an effective algorithm for real time dynamic object tracking, which uses only color features, so that the algorithm is sensitive to illumination and some other environmental elements. This paper presents and implements an effective moving object detection and tracking to reduce the influence of illumination interference, which improve the performance of tracking under similar color background. The implemented prototype system recognizes object using invariant features, and reduces the dimension of feature descriptor to rectify the problems. The experimental result shows that that the system is superior to the existing methods in processing time, and maintains better problem ratios in various environments.

**Key Words** : Object Detection, Object Tracking, Feature Descriptor, Feature Matching, Camshift, SURF

#### 1. 서 론

객체 검출과 추적 기술은 컴퓨터 비전 분야에서 광범위하게 활용되는 응용 기술이며, 군 관련, 차량 항법, 교통 통제 및 감시, 정보 감시와 같은 다양한 응용분야에서 활용되고 있다[1]. 특히, 최근 컴퓨터 비전 기술을 활용하여 객체를 검출하고 추적하는 응용 분야로 증강현실 서비스가 주목받고 있다[2]. 객체 검출 기술은 산업현장에서도 많이 활용되고 있다. 주로 컴퓨터와 카메라를 이용해 산업 현장에서 사람의 눈 역할을 하여 시각적으로 인식

하는 시스템은 머신 버전으로 특화되어 산업 기술에 반영되고 있다. 이미 반도체, 디스플레이, 자동차, PCB 등의 첨단 산업에서는 거의 모든 공정에 제품 검사를 위한 머신 비전 시스템이 설치되어 있다[17-18]. 또한 2017년 CES에서 LG전자에서 선보인 투명 디스플레이 장비는 많은 사람들의 이목을 집중시켰으며, 2018년부터 양산체제로 들어가는 투명 디스플레이 상에서 컴퓨터 비전의 기초 핵심 기술인 객체 검출은 다양한 형태의 애플리케이션을 배출할 수 있다. 이러한 객체 검출과 추적의 주된 목적은 비디오 시퀀스(Sequence)에서 객체의 이동 흐름과 동작을 획득하여 객체에 대한 다양한 형태의 분석 처리와 정보 제공이 가능하기 때문이다[3]. 산업 분야의 적용 가능성과

†E-mail: thinkthe@gntech.ac.kr

광범위한 활용성 측면에서 비디오/이미지를 기반한 컴퓨터 비전은 인간과 컴퓨터의 상호 작용에서 중요한 역할을 담당한다[4]. 통제 및 감시 시스템을 포함하여 객체 추적의 대표적인 응용 프로그램에서 효과적인 기법들이 제안되고 연구되고 있지만, 센서 잡음, 이동하는 객체의 외관 변화, 조명 변화, 복잡한 배경과 객체의 상호 작용 등 다양한 환경과 배경에 따라, 객체의 검출과 추적은 여전히 어려운 문제점을 가지고 있다[5]. 카메라로 입력 받는 비디오 시퀀스에서 대상 객체(Target Object)를 추적하는 대표적인 알고리즘으로, Bradscki가 칼라 히스토그램(Color Histogram)을 사용하여 목표 모델을 설정하고 이를 추적하는 Camshift (Continuously Adaptive Meanshift)가 있다[6-7]. 목표 모델의 히스토그램은 칼라 발생 확률 정보를 보존하기 때문에 대상 객체의 모양 변화에 쉽게 영향을 받지 않는 장점을 갖는다. 또한 그라디언트 디센트(Gradient descent) 방식을 사용하여 계산 시간을 단축함으로써, 대상 객체의 추적을 빠르게 한다는 특징이 있다. 이러한 Camshift 알고리즘을 통해 실시간 객체 추적이 가능하지만, 추적 대상 객체의 칼라 특성만을 사용하기 때문에, 다음과 같은 2가지 단점이 있다[8]. (1) 대상 객체와 배경이 유사한 칼라를 가질 때, 검출과 추적에서 오류가 발생할 가능성이 높으며, (2) 객체의 빠른 움직임에 따라, 동적으로 배경이 수시로 바뀔 때, 추적 대상 객체를 놓치는 손실이 발생할 수 있다. 다시 말해, Camshift는 조명과 배경 환경의 변화에 민감하다는 단점이 있다.

본 논문에서는 Camshift 알고리즘이 환경 변화에 적응성을 높이는 개선 방법을 제시한다. 제안 알고리즘은 배경을 포함하여 칼라 변화에도 강인한 객체 추적이 가능하고, 추적 과정에서 발생할 수 있는 추적 실패에 보다 빠르게 대처하기 위해 개선된 SURF 알고리즘을 적용하여 빠른 재탐색을 수행한다. 다양한 실험을 통하여, 제안 알고리즘이 대상 객체의 위치 검출과 추적에서 기존의 알고리즘보다 우수하다는 것을 평가 검증한다.

본 논문의 구성은 다음과 같다. 본 논문의 구성은 다음과 같다. 2장에서 객체 검출/추적과 관련된 연구동향과 본 논문의 기초 알고리즘으로 활용된 Camshift와 SURF 알고리즘에 대해 살펴보고 3장에서 성능 향상을 위한 개선 방법을 제시한다. 4장에서 실험 결과를 논의하고 5장에서 결론으로 마무리한다.

## 2. 관련연구

객체 추적에 사용되는 대표적인 추적 알고리즘은 Meanshift, Camshift, Kalman Filter 등이 있다[9]. 그 중 Meanshift

과 Camshift 알고리즘은 탐색 창을 통하여 추적 물체의 영역 및 중심을 계산한다[10-11]. Camshift는 Gary가 1988년에 처음으로 제안하였으며, Meanshift라는 확률 분포의 피크(Peak)를 찾기 위해 그라데이션(Gradient)을 동반한 비모수적(Non-parametric) 기법을 기반한다[12]. HSV 칼라 모델에서 색조(H)는 조명 영향을 받지 않기 때문에 칼라에서 색조값을 활용하여 연산한다[19]. 결국 채도(S)와 밝기(V)를 무시하기 때문에 문제가 발생될 수 있다. Camshift는 기본적인 이미지 처리 기법인 칼라 히스토그램을 대상 모델로 활용하며, Camshift의 실행절차는 다음과 같다.

(1) 계산 영역을 설정하기 위한 초기 탐색 창 크기와 위치를 결정한다. S와 V값의 임계값을 설정하고 해당 임계값 이하에서의 H 값만을 활용하며, 해당 영역의 칼라 히스토그램을 계산한다.

(2) 현재 이미지의 칼라 히스토그램의 색상 확률 분포 함수를 계산한다.

(3) 탐색 창 내에서 가장 큰 중심점을 찾고, 탐색 창을 중심으로 면적을 계산한다.

(4) 탐색 창이 수렴하면 (2)단계로 돌고, 그렇지 않을 경우는 (3)단계로 간다.

특징점 추출을 위한 알고리즘으로, 대표적으로 SIFT [13]과 SURF [14]이 있다. SIFT는 Lowe가 제안하였으며, 회전과 스케일에 영향을 받지 않는 이미지 특징 추출 알고리즘이다. 이미지 매칭 및 연결에서 많이 활용되지만 일반적으로 추출한 특징 정보와 계산 복잡도가 높다는 단점이 있다[20]. 이에 반해, SURF는 SIFT를 보완한 알고리즘으로 성능이 보다 우수하고 처리시간이 빠르다. SURF 특징 추출 연산은 키포인트(Keypoint) 검출, 기술자(Descriptor) 생성과 키포인트 매칭 과정으로 수행된다.

## 3. 제안하는 개선 알고리즘

### 3.1 Camshift 개선

전통적인 Camshift 방식은 H값만을 계산하며, S와 V에 대해서는 임계값을 설정하고 임계 범위 미만의 값을 유효한 값으로 판단한다. 이로 인해 객체 추적 중에 배경 또는 조명이 바뀌게 되면, 전통적인 Camshift 기법에서는 객체 추적에 실패할 확률이 높다[15]. 이는 배경 변화에 강인하지 못하다는 단점으로 나타나며, 추적 품질을 보장하지 못한다. 이를 해결하기 위해서는 서로 다른 조명에서는 유효 픽셀을 최대화하기 위해 초기화 단계에서 설정한 S와 V의 임계값을 동적으로 변경할 필요가 있다.

본 논문에서는 이러한 문제를 해결하기 위해, 조명이 강한 환경에서 높은 S값을 갖는 픽셀과 약한 조명 환경에

서 낮은 S값을 갖는 픽셀들을 선별적으로 제외시키고, 제한요소에서 탐색 창이 초기화된 다음 S와 V의 임계값을 각각 변경시킨다. 백-프로젝션(Back-projection) 비율이 전체 이미지와 검색 이미지 사이에서 최소값이 되며, 조정이 마무리된다. 탐색 창의 모든 픽셀에 대해 객체의 유효 픽셀 비율에 따라 결정되며, S와 V의 변화는 일정한 범위에서 객체가 검출되는 패턴을 보인다. 실제로, 특정 임계값보다 낮은 부분은 일반적인 잡음으로, 높은 부분에서는 객체의 불완전성을 갖는다. 본 논문에서는 실험적 측정에 따라 30% 비율에 맞춰, S와 V를 최소 30에서 최대 120사이로 설정하여 최상의 추적 성능을 얻는다.

### 3.2 SURF 개선

SURF 특징점 기술자는 모든 관점에서 대표성을 갖는 방향에 의해 결정되고 생성된다[16]. 대표적 방향을 계산하는 작업은 Haar 웨이블릿 응답에 기반을 하며, 관심점을 중심으로 반경  $6\sigma$ 인 원 영역의 x와 y좌표에서 Harr 응답을 계산한다. Harr 웨이블릿의 크기는  $4\sigma$ 이며, 벡터의 합은  $60$ 도마다 계산된다. 벡터의 합이 최대인 방향이 대표 방향으로 결정된다.

본 논문에서는 결정되는 대표 방향에 따라 클러스터링 필터를 사용하여 의미 있는 특징점만을 군집화하고, 군집되지 않는 특징점은 잡음(Noise)으로 간주하여 정합과정에서 특징점에서 제외시킨다. 특징점 개수를 감소시킴으로써, 불필요한 연산을 배제하여 처리시간의 단축 효과를 얻을 수 있다. 본 논문에서 적용하는 클러스터링 필터 내에 존재하는 특징점의 개수와 밀집도를 계산하고 특징점으로써의 가치를 판단하고 최종적으로 정합에 활용될 수 있는 특징점만을 생성한다. 이를 위한 특징점 분류는 수식 (1)과 같다.

$$s^2 = \frac{1}{N} \sum_{i=0}^{N-1} (X_i - \bar{X})^2 \tag{1}$$

$$G = \begin{cases} 1, & T_N < N \wedge T_s < s \\ 0, & otherwise \end{cases} \tag{2}$$

$s$ 는 클러스터의 밀집도를,  $N$ 은 각 클러스터에 포함된 특징점 개수를,  $X_i$ 는 특징점 좌표  $(x_i, y_i)$ 를,  $\bar{X}$ 는 특징점 클러스터의 중심 좌표를,  $T_N$ 은 클러스터링 필터내에 존재하는 특징점 개수에 대한 임계값을,  $T_s$ 는 클러스터링 필터 내에 존재하는 특징점들 간의 밀집도에 대한 임계값을 의미하며,  $G$ 는 특징점 클러스터링 필터를 통한 특징점 판별 여부를 나타낸다. 본 논문에서는 특징점 분류를 위한 클러스터링 필터 크기를  $21 \times 21$ 로 고정하였으며, 필터

내에 존재하는 특징점 개수는 전체 픽셀 수의  $\frac{1}{8}$ (12.5%) 수준으로 임계값을, 필터내 특징점간의 밀집도 임계값은 7로 설정하였다. 이는 다수의 실험을 통해 얻은 최적의 실험적 수치이다. 특징점 클러스터링 필터를 통해 군집화된 특징점을 분류하고, 의미 있는 특징점만을 선별하면 식 (3)과 같이 나타낼 수 있다.

$$F_c(x, y) = [f_1, f_2, \dots, f_n] \tag{3}$$

$F_c(x, y)$ 는 특징점 클러스터링 필터를 사용하여 의미 있는 특징점만을 분류한 집합을,  $f_n$ 은 선정된 특징점의 픽셀 위치를 나타낸다.

입력 이미지가 복잡하고 섬세할 경우, 이미지 엣지(Edge) 성분이 많이 추출되어, 특징점 개수가 많아진다. 이럴 경우, 클러스터링 필터 내부의 특징점 개수와 밀집도가 높아지며, 특징점 개수가 임계값보다 높으면 클러스터링 필터를 통한 분류가 연산을 가중시키는 결과를 얻는다. 이와 반대로, 입력 영상이 단순할 경우, 전체 특징점 개수가 적고 밀집도가 낮아지기 때문에 클러스터링 필터를 통한 연산에서 특징점이 많이 제거 되는 결과를 얻을 수도 있다. 다시 말해, 이미지의 복잡도에 따라 특징점이 많을 때와 적을 때에 대한 임계치를 적응적으로(Adaptively) 필터링을 해야 한다. 적응적 필터를 적용하기 위한 수식은 다음과 같다.

$$F_a = \begin{cases} 1, & T_{min} < img_f < T_{max} \\ 0, & otherwise \end{cases} \tag{4}$$

$img_f$ 는 이미지에서 추출된 특징점 개수를,  $T_{min}$ 과  $T_{max}$ 는 실험 데이터셋에 포함된 전체 이미지들에 대한 평균 특징점 개수 분포를 의미한다. 본 논문에서는 실험적 수치를 통해  $T_{min}$ 과  $T_{max}$ 를 하위 20%, 상위 20%를 적용하였다. 최종적으로, 적응적 클러스터링 필터를 통해 추출된 의미 있는 특징점을 생성하여 정합과정에 적용한다.

### 3.3 개선된 SURF를 적용한 Camshift 개선

색조(Hue) 특징만을 활용하여, 배경 변화에 민감한 Camshift 알고리즘의 단점을 보완하기 위해, 서로 다른 조명 변화에 견고하도록 자동적인 매개변수 조정 기능을 부여하여 Camshift알고리즘을 개선한다. 일반적으로, Camshift에서 객체 추적에 실패하는 경우는 2가지 상황에서 주로 발생한다. (1) 객체의 색조와 배경의 색조가 유사하여 혼동하는 경우와 (2) 객체가 빠르게 이동하여, 2개의 연속된 프레임 객체 영역이 오버랩되지 않아, 탐색 창이 이동속

도를 따라가지 못하는 경우이다. 이러한 문제를 해결하기 위해, 먼저 바타차랴 거리(Bhattacharyya Distance)를 계산하여 대상 객체와 배경간의 유사성을 계산한다.  $m$ 개의 칼라를 갖는 이미지에서 대상 객체에 대한 정규화된 칼라 다이어그램은  $p(u)(u = 1, 2, \dots, m)$ 이며, 배경에 대한 정규화된 칼라 다이어그램은  $q(u)(u = 1, 2, \dots, m)$ 이라고 하면, 바타차랴 거리는 식 (5)로 계산된다.

$$Bh(p, q) = \sum_{u=1}^m \sqrt{p(u)q(u)} \tag{5}$$

이상적으로, 배경 색상과 움직이는 객체의 색상이 완전히 같을 경우,  $Bh(p, q)$ 는 최대값으로 이르게 되지만, 실제적으로 그럴 가능성은 매우 희박하다. 실험에서 임계값을 0.8로 설정하였으며, 바타차랴 거리  $Bh$ 가 0.8 아래로 내려가면, 추적 중에 객체의 손실이 발생한 것으로 간주하여, 재추적을 수행한다. Algorithm 1은 이를 반영하여 개선한 Camshift의 단계를 표시한다.

**Algorithm 1.** Improved Camshift+SURF algorithm

```

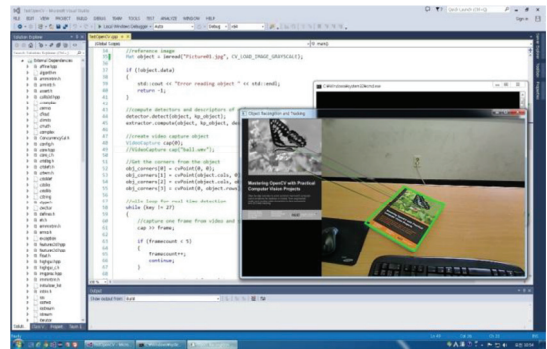
Input: Video sequences
Output: Location of the interest target object on the frame of video sequences
begin
while (Each frame of sequences) {
  Read the template image;
  Read 1st frame of sequence;
  Compute the location of target object using improved SURF;
  Compute color histogram of template image, and Calculate the threshold of S and V;
  while (next frame) {
    Update location and size of search Window using improved Camshift;
    Calculate Bhattacharyya distance;
    if (Bh < 0.8) {
      Decide to be lost the target, and re-compute the location using improved SURF;
    }
  }
}
end
    
```

객체 추적에 실패하였을 때, 개선한 SURF 알고리즘을 사용하여 전체 이미지에서 대상 객체와 일치하는 객체를 재탐색한다. 일치하는 관심 점을 발견하면, 관심 점의 중심을 계산하고 탐색 창 주변에 점들을 대상으로 Camshift 알고리즘을 통해 계속적으로 객체 추적을 수행한다. 이를

통해 실시간으로 객체를 추적하고, 객체 칼라와 유사한 배경에서도 강인하게 추적이 가능하다.

**4. 실험 및 평가**

개발 환경으로, Visual Studio 2015에서 C++을 사용하여 구현하였으며, 오픈 라이브러리인 OpenCV를 사용하였다. 실험은 Intel Core™ 2.7GHz, 8GB 메모리가 장착된 PC에서 실행되었다. Fig.1은 개발 시스템에 대한 예시를 보여준다.



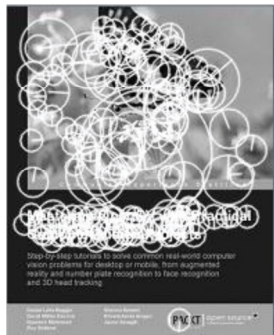
**Fig. 1.** Screenshot of the implemented system; Target object and tracking result.



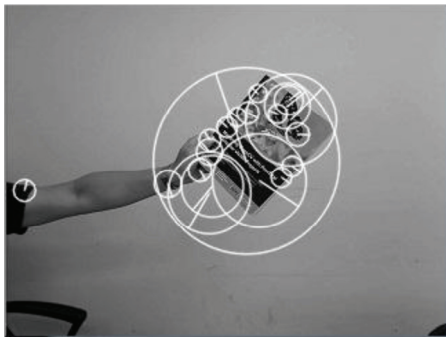
**Fig. 2.** Result of object detection and tracking with implemented system based on improved Camshift+SURF.



본 논문에서 제시하는 개선된 Camshift+SURF 알고리즘을 실험으로 측정 평가하며, 다수의 입력 비디오 시퀀스에 대해 움직이는 객체를 검출하고 추적 실험을 수행한다. 전체 프레임 대비, 탐색 창에 대해 객체의 외곽선을 포함하여 객체 영역의 90% 이상인 프레임의 비율을 성공으로 간주하며, 실시간 테스트를 위해 웹 카메라에서의 입력을 대상을 수행한다. Fig. 2는 구현한 프로토타입 시스템의 인터페이스를 보여주며, Fig. 3은 실시간 환경에서 구현한 시스템을 통해 객체를 검출하고 추적한 결과 화면을 보여준다.



(a) Screenshot of the extracted feature points on the target object



(b) Screenshot of the interest object tracking

**Fig. 3.** Result of feature extraction with implemented system based on improved SURF.

### 5. 결론

본 연구에서는 Camshift와 SURF를 통해 움직이는 객체를 검출하고 추적하는 알고리즘을 살펴보고, 빠른 객체의 움직임과 배경 화면 변화에 강한 추적 알고리즘을 구현하였다. 본 논문에서 제시한 개선된 Camshift+SURF 알고리즘은 실시간 환경에서 동작하기 위해 추적 대상 객체에 대한 추출 특징점의 차원을 감소시키며, 추적 실패

시에 재탐색을 보다 용이하게 지원함을 실험을 통해 검증하였다. 본 논문의 주된 학술적 공헌으로는, 낮은 차원의 특징점 추출과 추적 실패에서의 빠른 재탐색 기능을 제공하는 알고리즘을 구현하였다는 것이 특징이며, 개선된 특징점 추출을 통해 증강현실/가상현실 객체 인식과 감시 시스템 등 다양한 응용 프로그램으로의 적용성 실험을 향후 연구 내용으로 남긴다.

또한 차세대 디스플레이 장비로 주목 받는 투명 디스플레이는 객체 검출과 추적의 기초 기술로써 다양한 응용분야가 발생할 것으로 예측되며, 투명 디스플레이 상에서 대상 객체를 검출하고 움직임을 추적하는 형태의 기초 학습으로 활용될 것이다.

### 감사의 글

이 논문은 2017년도 경남과학기술대학교 대학회계 연구비 지원에 의하여 연구되었음.

### 참고문헌

1. Yang H., Shao L., Zheng F., Wang L. and Song Z., "Recent Advances and Trends in Visual Tracking: A Review", *Neuro-computing*, 74(18), pp. 3823-3831, (2011).
2. Yilmaz A., Javed O. and Shah M., "Object Tracking: A Survey", *ACM Computing Surveys*, 38(4), (2006).
3. Klohofer W. and Kampel M., "Interest Point based Tracking", *International Conference on Pattern Recognition*, pp. 3549-3552, (2010).
4. Angela Zhou, "Cybernetics and Human-Computer Interaction: Case Studies of Modern Interface Design", *International Conference on Multidisciplinary in IT and Communication Science and Applications*, pp. 1-6, (2016).
5. Yue Y., Gao Y. and Zhang X., "An Improved Camshift Algorithm Based on Dynamic Background", *International Conference on Information Science and Engineering*, pp. 1141-1144, (2009).
6. Zhang J. Y., Wu H. Y., Chen S. and Xia D. S., "The Target Tracking Method based on Camshift Algorithm combined with SIFT", *Advanced Materials Research*, 186(1), pp. 281-286, (2011).
7. Exner D., Bruns E., Kurz D., Grundhöfer A. and Bimber O., "Fast and Robust Camshift Tracking", *International Conference on Computer Vision and Pattern Recognition*, pp. 9-16, (2010).

8. Aomei Li, Wanli Jiang, Weihua Yuan, Dehui Dai, Siyu Zhang and Zhe Wei, "An Improved FAST+SURF Fast Matching Algorithm", International Congress of Information and Communication Technology, pp. 306-312, (2017).
  9. Wang J., He F., Zhang X. and Gao Y., "Tracking Objects through Occlusions using Improved Kalman Filter", International Conference on Advanced Computer Control, 5(1), pp. 223-228, (2010).
  10. Leichter I., Lindenbaum M. and Rivlin E., "Meanshift Tracking with Multiple Reference Color Histograms", Computer Vision and Image Understanding, 114(3), pp. 400-408, (2010).
  11. Ahn H., Lee Y., Lee J. and Cho H., "Research on Target Tracking based on CamShift Approach with Feature Matching", International Conference on Convergence Technology, pp. 930-931, (2015).
  12. G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," Intel Technology Journal, 2nd Quarter, (1998).
  13. Lowe D. G., "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, 60(2), pp. 91-110, (2004).
  14. Bay H., Tuytelaars T. and Van Gool L., "SURF: Speeded-Up Robust Features", International Conference on ECCV, pp. 404-417, (2006).
  15. Viola P. and Jones M., "Robust Real-time Object Detection", International Journal of Computer Vision, 4(1), (2001).
  16. Juan L. and Gwun O., "A comparison of SIFT, PCA-SIFT and SURF", International Journal of Image Processing, 3(4), pp. 143-152, (2009).
  17. Marcin Kuzanski, Anna Fabijanska, Dominik Sankowski, "Machine Vision – Automation of Selected Measurement Systems", International Conference on Perspective Technologies and Methods in MEMS Design, (2008).
  18. Wai Lee, "3D Machine Vision in IoT for Factory and Building Automation", International Symposium on Circuits and Systems, (2017)
  19. Woori Han, Youngseop Kim, Yong-Hwan Lee, "Multi-Object Tracking based on Keypoints", Journal of the Semiconductor and Display Technology, 14(3), pp. 67-72, (2015)
  20. Yong-Hwan Lee, Je-Ho Park, Youngseop Kim, "Comparative Analysis of the Performance of SIFT and SURF", Journal of the Semiconductor and Display Technology, 12(3), pp. 59-64, (2013)
- 
- 접수일: 2017년 12월 27일, 심사일: 2017년 12월 27일,  
 게재확정일: 2017년 12월 27일