

# 색상변화를 갖는 객체추적 알고리즘

황인택<sup>†</sup>, 최광남<sup>\*\*</sup>

## 요 약

기존의 색상 기반의 Mean Shift 알고리즘을 이용한 객체추적 알고리즘은 초기 색상 정보가 사라질 경우 정확한 객체추적을 수행할 수 없다. 본 논문은 객체의 색상이 변할 때 색상 정보를 변경하여 정확히 추적하는 알고리즘을 제안한다. 제안 알고리즘은 현재의 위치를 중심으로 다음 객체 위치에 해당하는 밀도가 가장 높은 위치를 Mean Shift 알고리즘으로 구하고, 바뀐 색상 정보를 변경하는 반복적인 기법을 사용한다. 이를 통해 처음 설정한 객체의 색상이 바뀌거나 사라지더라도 정확한 객체추적을 할 수 있게 되었다. 본 논문에서는 제안 알고리즘을 구현하고, 실험 결과로 성능을 입증한다.

## An Algorithm for Color Object Tracking

Inteck Whoang<sup>†</sup>, Kwang Nam Choi<sup>\*\*</sup>

## ABSTRACT

Conventional color-based object tracking using Mean Shift algorithm does not provide appropriate result when initial color distribution disappears. In this paper we propose a tracking algorithm that updates the object color sample when the color is changing. Mean Shift analysis is first used to derive the object candidate with maximum increase in density direction from current position. The color information of object is updated iteratively. The proposed algorithm achieves accurate tracking of objects when initial color samples are changed and finally disappeared. The validity of the effective approach is illustrated by the experimental results.

**Key words:** Object Tracking(객체추적), Mean Shift(색상변화), Radical Color Changes(중심점 이동)

## 1. 서 론

실시간 객체추적(Tracking)은 비디오 영상에서 하나의 영상(Frame)과 다른 영상 사이의 특정 객체를 찾거나 추적 대상의 행동이나 특징을 감시하는 등의 여러 분야에서 두루 사용되고 있다. 일반적으로 비디오 영상의 지정된 하나의 영상을 기준으로 이웃한 두 개의 순차적인 영상들 간의 시간은 아주 짧고 대부분 일정하다. 그러므로 두 영상간의 변화는 상당히 제한적으로 나타난다. 영상 간의 한정된 변화는

영상에 포함된 객체 특징(Feature) 추출 후 비교해서 알아낼 수 있다. 그러나 이 특징을 추출하기 위해서는 다양한 조명상태와 시점의 변화, 복잡한 배경, 대상 객체의 크기변화와 다른 사물에 의한 가려짐(Occlusion) 등을 고려해야한다. 그러므로 다양한 외부변화로 발생하는 문제점들을 해결할 수 있는 강한 객체 특징 추출 알고리즘이 필요하다. 또한 대상에 따라 객체 형태가 고정되지 않는 내부적인 경우까지 고려해야 한다. 위에서 언급한 바와 같이 객체추적을 위해서는 객체 특징 추출이 반드시 필요하다. 이 과

\* 교신저자(Corresponding Author): 최광남, 주소: 서울시 동작구 흑석동 221(156-756), 전화: 02)825-5316, FAX: 02)820-5301, E-mail: knchoi@cau.ac.kr

접수일: 2006년 12월 21일, 완료일: 2007년 5월 29일

<sup>†</sup> 준회원, 중앙대학교 컴퓨터공학부

(E-mail: sitdown@vim.cau.ac.kr)

<sup>\*\*</sup> 정회원, 중앙대학교 컴퓨터공학부

\* 이 논문은 2003년도 중앙대학교 학술연구비 지원에 의한 것임.

정에서 내외적 환경을 고려해서 불필요한 정보를 제거하고 우리가 원하는 추적 대상의 특징만을 유지해야 한다. 이것이 객체추적 분야의 가장 중요한 초점이다.

객체추적 시스템은 여러 형태의 잡음과 관찰데이터의 변화에 관계없이 수행되어야 한다. 일반적으로 이 조건을 만족시키기 위해 사전 지식을 추적 알고리즘에 반영하여 해결한다. 그러나 대상 혹은 상황에 따라서 상당히 곤란한 경우가 종종 발생한다. 그 중 한 예가 대상 객체의 색상변화이다. 색상 정보를 기반으로 하는 대부분의 알고리즘들은 객체가 추적되는 동안 객체의 색상이 변하지 않는다는 것을 전제로 한다. 그러나 우리는 일상생활에서 이 전제를 벗어나는 예를 쉽게 찾을 수 있다. 옷을 갈아입거나, 고개를 뒤로 돌리거나, 좌우 다른 색을 가진 컵 등이 그 예이다. 모두가 동일한 객체이지만 시점이나 행위자의 행동에 따라서 초기에 보였던 색상과 다른 색상을 나타내기 때문에 다른 대상으로 인식될 수 있다. 그러므로 색상기반의 추적방법은 형태가 변할 수 있는 객체추적에 좋은 성능을 발휘하지만, 큰 색상변화를 가지는 객체에 대해서는 추적이 어렵다는 문제점을 안고 있다.

Comaniciu는 Mean Shift 알고리즘을 이용하여 객체를 추적하는 방법을 제안하였다[1]. Comaniciu의 논문에서 제시한 실험결과 중 객체의 색상이 변하는 예가 있다. 그러나 상세히 분석해보면 이 역시 초기에 설정한 색상이 객체추적에 필요한 최소치를 유지함으로써 이루어짐을 알 수 있다. 만약 추적 대상이 초기에 설정한 색상이 사라진다면 Mean Shift 알고리즘 그대로 적용하는 것은 힘들다.

본 논문에서는 기존의 색상 기반 객체추적 알고리즘이 해결하지 못한 큰 색상의 변화를 가지는 객체의 추적방법을 제안하고자 한다. 본 논문은 다음과 같이 구성되어 있다. 2장에서 객체추적 알고리즘에 관련된 연구를 간략히 소개한다. 3장에서는 색상기반의 Mean Shift에 관하여 제안한 알고리즘에 대해 상세히 설명한다. 4장에서는 3장에서 제안한 알고리즘과 기존 Mean Shift 알고리즘을 이용하여 실험하고 그 결과를 비교 분석한다. 마지막 5장에서는 실험결과를 바탕으로 결론을 기술하고 향후 연구방향에 대해서 논의한다.

## 2. 관련 연구

객체추적을 위해 현재까지 다양한 방법이 시도되었다. 대표적으로 템플릿(Template)기반 방법, 칼만 필터(Kalman Filter)를 이용한 방법, Mean Shift 알고리즘을 이용한 방법 등이 있다. 템플릿을 이용한 객체추적방법은 기준이 되는 영상에서 객체의 외형적인 특징을 추출하고, 다음 영상에서 같은 방법으로 추출한 특징들과 비교하여 추적하는 방법이다. 최근 Iain Matthews는 기존 템플릿(Template)기반 방법들이 가지는 추적오류(Drifting) 문제를 해결하기 위해 자가학습적인(Heuristic) 갱신(Update) 알고리즘을 제시하였다[2]. 그리고 이 방법은 일반적인 동적 외관모델(Active Appearance Model : AAM)을 특정 모델에 적합하도록 확장되었다. 그러나 템플릿 기반 방법은 객체의 외관을 대상으로 특징을 추출하기 때문에 Mean Shift 알고리즘과는 달리 객체의 외관이 일정 이상 변하는 경우에는 추적이 어렵다는 단점이 있다.

칼만 필터는 물리적인 모델을 기반으로 객체의 다음 상태를 예측하고, 현재 상태를 측정해서 다시 객체의 상태를 나타내는 동적 방정식(Dynamic Equation)의 인자에 반영하는 반복적인 갱신구조를 가진다. Boykov와 Huttenlocher는 자동차를 대상으로 칼만 필터를 이용하여 추적 시스템을 개발하였다[3]. 객체의 상태에 대해서 어떠한 가정도 하지 않는 Mean Shift 알고리즘에 비해 칼만 필터는 객체의 상태를 선형적 방정식으로 표현해야 하는 단점을 가지고 있다.

이 외에도 McKenna는 가우시안 혼합 모델(Gaussian Mixture Model)을 이용하여 조명과 시점의 변화에도 강한 추적 시스템을 개발하였다[4]. Jepson은 외관모델(Appearance Model)의 추정을 위해 온라인 EM(Expectation Maximization) 알고리즘을 사용하여 가려짐이나 실세계의 객체의 외관 변화에도 강한 추적 시스템을 개발하였다[5]. 이 방법은 초기의 측정값과 관찰된 데이터의 우도값(Likelihood)의 반복적인 갱신을 통해서 주어진 관찰 데이터와 현재 매개변수 측정을 통해 완전한 데이터일 경우 예상되는 우도값(Log Likelihood)을 예측해 추적을 수행하였다. 이 경우 가우시안 혼합 모델의 가우시안 개수를 미리 결정해야 한다. Gordon과

Nummiaro는 베이지안(Bayesian) Bootstrap Filter와 파티클 필터(Particle Filter)를 이용하여 객체의 점침과 회전 및 크기 변화에 강한 추적 알고리즘을 제시하기도 하였다[6,7].

Mean Shift 알고리즘은 통계적으로 해당 모델의 각 샘플이 속한 곳으로 이동하는 비모수(Non-parametric) 알고리즘이다[8]. Mean Shift 알고리즘은 K-means Clustering 알고리즘과 마찬가지로 일반화되어질 수 있다[9]. 초기 Mean Shift가 발표될 당시에는 데이터 Clustering을 위해 개발되었지만 차후 여러 분야에서 광범위하게 사용되어 영상검색 및 동작 추적에 활용하거나 신체 추적에 응용되었다[10-12]. Babu는 Mean Shift 알고리즘과 칼만 필터를 결합하여 감시 및 모니터링이 가능한 틀을 제안하였다[13]. Meer는 영상의 중요한 특징들 복원하는 방법으로 Mean Shift 알고리즘을 다른 곳에 적용하였다[14]. Comanicu는 형태가 자유로운 객체추적에 활용하였다. 그러나 객체가 항상 같은 색상을 가진다는 전제가 필요했다[10].

### 3. 개선된 Mean Shift 알고리즘

Mean Shift 알고리즘은 커널밀도추정을 기반으로 밀도가 가장 높은 위치를 검색한다. 최고 밀도 위치를 찾기 위해 Mean Shift 벡터를 활용하여 반복적으로 최고점을 향하여 이동한다. Mean Shift 벡터는 현재 위치의 밀도와 주변의 최고 밀도와의 차이를 이용해 계산한다. 밀도를 구하기 위해 Mean Shift는 색상 정보를 활용한다. 이 때 추적 대상의 형태 정보는 사용하지 않는다. 색상 정보만을 이용하기 때문에 다양한 모양을 가진 객체를 추적할 때 다른 알고리즘에 비해 효과적으로 활용된다.

#### 3.1 Mean Shift 알고리즘을 이용한 객체추적

그림 1은 Mean Shift 알고리즘을 활용하여 밀도 최고점을 탐색하는 추적 과정을 보여주고 있다. 그림 1의 (a)와 (b)에 표시된 동그란 작은 점들은 해당 점의 위치에서 일어난 사건을 나타낸다. 표시된 모든 점은 서로 독립으로 가정한다. 만약 사건이 일어날 확률이 가장 높은 위치를 알고 싶을 경우 우리는 Mean Shift 알고리즘을 활용할 수 있다. 사건이 일어날 확률이 가장 높은 곳은 사건이 가장 많이 발생한

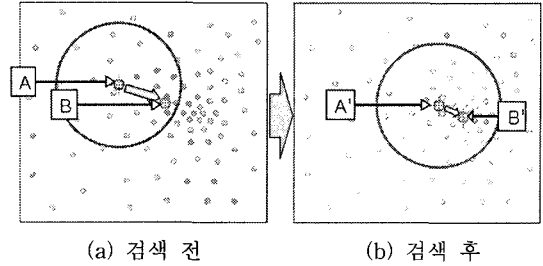


그림 1. Mean Shift 알고리즘을 이용한 지역 밀도 최고점(Local density maximum) 탐색 과정

위치 주변이 될 것이며, 이는 가장 밀도가 높은 위치이다. 그러므로 Mean Shift를 이용하여 해당 사건이 일어날 가장 높은 확률을 가진 위치를 쉽게 찾을 수 있다.

그림 2-(a)에서 A로 표시된 원의 중심지점을 초기 시작 위치로 가정한다면, 시작 위치를 중심으로 (a)의 큰 원의 크기에 해당하는 범위만큼 주변 밀도를 검색한다. 밀도는 원의 내부에 속하는 점들의 개수로 대신할 수 있다. 검색 결과 그림 2-(a)의 B 위치가 가장 높은 밀도 위치로 검색되었다면, 현재 그림 1-(a) A의 위치의 밀도 값과 B의 값을 비교해서 B가 더 높을 경우 위치를 이동하게 된다. 이동하는 방향과 거리는 A와 B간의 밀도 차이를 이용해 계산하게 된다. 위치가 이동된 모습을 그림 2-(b)에서 보여주고 있다. 그림 2-(a)에서 B 위치는 그림 2-(b) A'로 갱신된다. 이후 다시 이전 작업과 동일하게 주변의 밀도를 검색한다. 주변 검색 결과 B'가 가장 높은 위치로 나왔다면 다시 A'와 B'의 밀도 비교 후 이동을 결정하게 된다. 이 과정을 갱신된 위치의 밀도가 주변 밀도보다 가장 높게 나타날 때까지 반복하게 된다.

Mean Shift 알고리즘을 객체추적에 적용할 경우 그림 2의 각각의 점들은 추적 객체의 밀도함수로 표현되는 특징점이 될 수 있으며, A와 A'는 추정된 객체의 위치가 되고, B와 B'는 검색된 지역 밀도 최고점으로 예상되는 다음 객체 위치가 된다.

Mean Shift 알고리즘을 객체추적에 활용하기 위해서 본 실험에 적합한 형태의 커널 함수를 선택한다. 본 논문에서는 커널 함수로 Comanicu가 제안한 방법에 따라, 확률밀도함수를 부드럽게 하는 방사대칭커널을 사용한다[15]. d개의 차원을 가지는 다차원 공간  $R^d$  안에서 n개의  $\{X_i\}_{i=1, \dots, n}$  집합이 있다고 가정

한다면,  $x$ 의 밀도는 아래의 수식으로 계산할 수 있다. 단,  $K$ 는 커널함수를 나타낸다.

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (1)$$

여기서 커널 함수  $K$ 는 아래의 네 가지 조건을 만족하는 함수여야 한다.

$$\int_{R^d} K(x)dx = 1, \int_{R^d} xK(x)dx = 0, \lim_{\|x\| \rightarrow \infty} \|x\|^d K(x) = 0, \int_{R^d} xx^T K(x)dx = cI \quad (2)$$

$\|\cdot\|$ 는 유클리드 공간의 Norm,  $c$ 는 상수값,  $h$ 는 window radius로 가정한다.

이 중 가장 일반적인 커널로서 Epanechnikov kernel과 정규분포가 있다[16]. 본 논문에서는 일반적인 문제에 대해 가장 많이 활용되는 Epanechnikov kernel을 사용하였다. 본 논문에서 사용한 Epanechnikov kernel 커널 수식은 아래와 같다.

$$K_E(x) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|x\|^2) & \text{if } \|x\| < 1, \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$c_d$  : the volume of the unit sphere in  $R^d$

우리는 커널밀도 추정값의 변화도로 밀도변화를 측정할 수 있다.

$$\hat{\nabla}f(x) \equiv \nabla\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n (x-x_i) \nabla K\left(\frac{x-x_i}{h}\right) \quad (4)$$

( $h$  : window radius)

$$\hat{\nabla}f_E(x) = \frac{1}{nh^d} \frac{d+2}{h^2} \sum_{x_i \in S_h(x)} (x_i - x) = \frac{n_x}{n(h^d c_d)} \frac{d+2}{h^2} \left[ \frac{1}{n_x} \sum_{x_i \in S_h(x)} (x_i - x) \right] \quad (5)$$

그러므로 식(4)에서  $x$ 에 대한 확률밀도함수를  $f(x)$ 라고 가정한다면, 밀도의 변화도는  $\hat{\nabla}f(x)$  통해 추정할 수 있다. 이 값은  $\nabla\hat{f}(x)$ 와 근사값을 가진다. 이는 다시 커널 함수의 변화도  $\nabla K\left(\frac{x-x_i}{h}\right)$ 를 추정함으로써 알아낼 수 있다. 식 (5)는 Epanechnikov 커널 함수를 식 (4)에 대입하여 나온 식이다. 식 (5)에서  $S_h(x)$ 는  $n_x$  데이터를 포함하며  $x$ 를 중심으로 하는

$h^d c_d$ 의 크기와 반경  $h$ 를 가지는 공간이다. 식 (5)의 마지막 괄호에 해당하는  $\left[ \frac{1}{n_x} \sum_{x_i \in S_h(x)} (x_i - x) \right]$ 부분을 가리켜 *Sample Mean Shift* 라고 하며 [17],

$$M_{h,U}(x) \equiv \frac{1}{n_x} \sum_{x_i \in S_h(x)} (x_i - x) = \frac{1}{n_x} \sum_{x_i \in S_h(x)} x_i - x \quad (6)$$

( $S_h(x)$  : hypersphere of radius  $h$ )

위의 식 (6)과 같이 정리할 수 있다. *Sample Mean Shift*는 (6)의 식에서 Epanechnikov 커널 함수의 차를 이용해 계산할 수 있다. 그러므로 우리는 식 (5)의  $\left[ \frac{1}{n_x} \sum_{x_i \in S_h(x)} (x_i - x) \right]$ 을 다음과 같이 대체할 수 있다.

$$\hat{\nabla}f_E(x) = \hat{f}_U(x) \frac{d+2}{h^2} M_{h,U}(x) \quad (7)$$

위 (7)식과 같이 바꿀 수 있고, 다시  $M_{h,U}(x)$ 을 좌항으로 하는 식(8)이 성립할 수 있다.

$$M_{h,U}(x) = \frac{h^2}{d+2} \frac{\hat{\nabla}f_E(x)}{\hat{f}_U(x)} \quad (8)$$

그러므로 우리는 식 (8)에서와 같이  $x$  중심의 추정된 Uniform kernel의 밀도함수  $\hat{f}_U(x)$ 를 가지고 *Sample Mean Shift*에 해당하는  $\left[ \frac{1}{n_x} \sum_{x_i \in S_h(x)} (x_i - x) \right]$ 부분을 계산함으로써  $\hat{\nabla}f_E(x)$ 값을 구할 수 있다. 식 (8)의  $M_{h,U}(x)$ 를 가리켜 Mean Shift 벡터(vector)라고 하며, 이 값을 통해 이동방향과 이동거리를 알게 된다. 즉 이동방향과 이동거리는 평균값이 0인  $x$  위치를 중심으로 Epanechnikov 커널 함수로 변환된 값의 차이를 계산해서 결정하게 된다. 이를 바탕으로 이동할 위치를 반복적으로 계산할 수 있음을 알 수 있다. 우리는 현재 위치에서부터 Mean Shift 벡터를 통해 가장 큰 밀도 증가를 보이는 방향을 위의 수식을 통해 얻을 수 있다. 그 방향을 따라 새로운 위치로 이동한다. 이 과정을 지역밀도가 증가하는 위치로 이동해서 수렴할 때까지 이 과정을 반복하게 된다. 이때 초기 위치는 지역 밀도가 최대가 되는 지점으로 이동하게 된다. 이런 과정을 통해서 Mean Shift 알고리즘이 동작한다. Mean Shift 알고리즘은 이처럼 반복적인 Mean Shift 벡터의 계산이 요구된다. Mean

Shift 벡터의 계산 시간이 항상 수렴하여 유한한 시간 내에 계산이 끝남을 증명하기 위해  $\hat{\nu}f(x)$ 이 수렴함을 증명하여야 하는데 이는 Comanicu의 증명으로 대신한다[18].

### 3.2 제안 알고리즘

기존의 Mean Shift 알고리즘을 이용한 추적 알고리즘은 비정형적인 객체가 고정된 색상을 가진 경우에 적용되어 왔다. 그러나 기존의 방법은 추적 대상의 색상이 변할 경우에 그 기능을 제대로 수행하지 못하고 에러를 발생하게 된다. 이 때 색상의 변화를 감지할 수 있다면 추적 대상의 색상 정보를 갱신함으로써 이 문제를 해결할 수 있다. 그러나 추적 대상이 다양한 형태를 가질 경우 배경과 대상의 구분이 난해하기 때문에, 색상변화를 알아내기란 쉽지 않다. 본 논문에서는 개선된 Mean Shift 알고리즘을 통해 이 문제를 해결하고자 한다.

**step 1 :** 초기 추적 대상에 대한 밀도함수를 검색창(Search Window)으로 설정하고, Mean Shift 알고리즘 사용하기 위한 검색창의 크기  $h$ 를 설정한다.

**step 2 :** 다음 영상을 가져온다.

**step 3 :** 현재 위치를 중심으로 Mean Shift 알고리즘을 이용하여 다음 영상에서 객체의 위치를 얻는다.

**step 4 :** 새로운 위치를 중심으로 밀도함수를 구하고 색상변화도를 구한다.

**step 5 :** 변화도가 임계값 이상일 경우 새로운 밀도함수로 검색창을 갱신한 후 2단계로 이동, 아니라면 곧바로 2단계로 이동한다.

첫 번째 단계에서는 검색을 위한 초기 설정을 하게 된다. 검색창의 반지름 크기  $h$ 는 그림 1에서의 큰원의 반지름에 해당한다. 이 값은 실제 객체추적을 구현할 경우 성능에 가장 큰 영향을 주는 요소이다.  $h$ 값이 커지면 검색 범위가 넓어져서 정확성은 높아지지만 이를 계산하기 위한 시간이 많이 소비하게 되고, 줄이면 계산 시간은 줄어드는 반면 정확성은 떨어질 수 있다. 다음으로 Mean Shift 알고리즘을 사용하기 위해 밀도함수를 구한다. 본 논문에서는 밀도함수를 구하기 위해 컬러 영상의 적색, 녹색, 파란색(R,G,B)의 3개 색상채널에 대한 히스토그램을 사용하였다. 이를 기반으로 3차원의 밀도함수를 구한

다. 제안한 알고리즘의 첫 번째 단계에서 세 번째 단계까지는 기존 Mean Shift 알고리즘의 과정과 거의 흡사하다.

그러나 제안 알고리즘은 기존의 Mean Shift 알고리즘에 4단계와 5단계를 추가한다. 4단계에서 색상변화도를 구하고 5단계에서 검색창을 갱신하는 과정이 추가된다. 색상변화도는 객체의 색상변화를 나타내는 값으로 이 값을 통해 객체의 색상변화를 감지할 수 있다. 만약 색상변화도가 임계값 이상일 경우 검색창의 정보를 갱신한다. 이 과정을 반복함으로써 제안한 알고리즘은 추적 객체의 색상이 변하더라도 정확한 추적을 할 수 있다. 색상변화도는 아래의 식을 이용하여구한다.

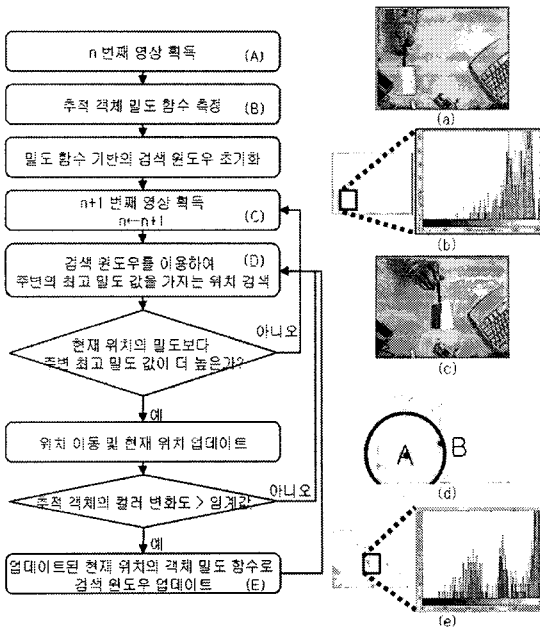
$$\nabla C = \sum |f_{rgb}(x) - f_{rgb}(\hat{x})| \tag{9}$$

식 (9)에서  $\nabla C$ 는 색상변화도,  $f_{rgb}$ 는 RGB 3채널의 히스토그램 함수,  $x$ 와  $\hat{x}$ 는 현재의 객체와 이전 단계에서 획득된 객체를 의미한다. 임계값은 객체의 색상변화 유무를 판별하는 기준으로 값이 작을수록 객체의 색상변화를 잘 반영하지만 노이즈에 영향을 많이 받을 수 있다. 반대로 값이 클수록 노이즈에 영향을 덜 받지만 색상변화를 잘 반영하지 못한다. 그러므로 외부 환경과 추적 객체에 따라 적절한 임계값을 설정하여야 한다.

보다 상세한 설명을 위하여 그림 2는 제안된 알고리즘의 상세한 흐름도와 그림 예제를 함께 보여준다. 그림 2의 왼쪽 (가)는 제안한 알고리즘의 상세한 흐름도를 보여주고 오른쪽 그림 예제 (나)는 (가)의 영어 대문자에 해당하는 예제 그림을 보여준다. 그림 2-(a)와 같이 추적 객체를 포함하는  $n$ 번째 입력 영상을 획득한다. 다음으로 정의된 크기의 검색창으로 추적 객체를 선정 후, 중심점을 설정하고 밀도함수를 구한다. 그림 2-(b)는 밀도함수의 예로써 명암을 1차원의 히스토그램으로 나타내었다. 초기 색상이 밝은 색(노란색)이므로 밝은 명암값을 가지는  $x$ 축의 오른쪽으로 히스토그램의 막대가 몰려있음을 볼 수 있다.

이 작업이 끝나면  $n+1$ 번째 영상을 획득하고 이전 영상에서 설정한 위치를 중심으로 주변 밀도를 측정한다. 그림 2-(d)는 주변 밀도측정에 해당하는 예제를 보여주고 있다. 이전 영상인 2-(a)에 나타난 객체의 위치를 2-(d)에서 A로 표시한다. 이전 객체 위치

인 A를 중심으로 검색 범위를 설정한다. 검색 범위는 2-(d)에서 A가 중심인 원부분에 해당한다. 원 중심에서 모든 방향으로 주변의 밀도함수를 구한다. 다음으로 이전 영상과 비교하여 현재 영상에서 객체의 위치를 추정한다. 객체의 위치는 객체가 나타날 확률이 가장 높은 곳 즉 가장 높은 밀도를 가진 위치와 동일하므로 그림 2-(d)의 추적 대상의 중심위치와 동일한 그림 2-(d)의 B가 선정되었음을 알 수 있다. 다음으로 이전 객체 위치 A와 현재 추정된 객체 위치 B의 밀도를 비교한다. 만약 A가 더 높은 밀도를 가진다면 객체는 위치이동을 하지 않았으므로 곧바로 다음 영상을 입력 받는다. 그림 2-(d)에서는 B가 A보다 더 높은 밀도함수를 가지기 때문에 B의 위치로 이동한다. 이동 후 B 위치를 추정된 객체의 중심점으로 다시 설정하고, 이전 객체 밀도함수인 그림 2-(b)와 새롭게 선정된 B 위치의 밀도함수 그림 2-(e)간의 색상변화도를 측정한다. 만약 색상변화도가 정의된 임계값 보다 큰 경우에는 그림 2-(e)의 밀도함수로 갱신된다. 이후 다시 다음 영상을 획득하고 다시 주변 밀도 측정과 위치이동을 반복하게 된다. 제안 알고리즘은 위에 언급한 과정을 입력 영상에 대해 반복적으로 수행하여 객체를 추적한다.



(가) 흐름도 (나) 그림 예제

그림 2. 제안한 알고리즘의 상세한 흐름도와 그림 예제

#### 4. 실험결과 및 분석

본 논문의 실험에서는 PC용 카메라인 Logitech 회사의 QuickCam Pro 4000을 이용하였으며, 해상도는 320\*200 24bit 색상이며, 초당 15장의 영상을 전송받는다. 제안 알고리즘의 성능 입증에 위하여 모든 실험은 동일한 실내조명과 동일한 객체를 바탕으로 기존 Mean Shift 알고리즘과 제안 알고리즘을 서로 비교한다. 본 논문에서는 실험대상으로 제안 알고리즘을 명백하게 보여줄 수 있는 인공객체와 실제계에서 사용하는 컵과 이동하는 사람을 선정하였다. 추적 대상은 모두 기존의 Mean Shift 알고리즘을 이용한 객체추적 시스템과의 성능 비교를 위하여 색상 공간에서 거리가 먼 서로 다른 두 가지의 색상을 가지는 객체로 선정하였다. 모든 실험대상은 위치이동과 동시에 색상변화를 매 영상별로 가지며 최종 영상에서는 초기에 설정한 객체 색상이 완전히 사라지게 된다. 모든 실험에서 추적 객체가 위치이동과 색상변화를 동시에 함으로써 제안 알고리즘이 기존의 Mean Shift 추적 알고리즘 성능을 유지하면서 색상변화에도 강하다는 것을 입증한다.

그림 3은 기존 Mean Shift 알고리즘을 이용한 객체추적과 제안 알고리즘의 차이를 가장 잘 보여줄 수 있는 간단한 객체에 대한 실험 결과이다. 추적 객체는 초기에는 밝은색(노란색)만을 가지고 있으며 이후 위치이동과 동시에 밝은색은 차츰 어두운색(보라색)으로 변하게 되고 최종적으로 초기의 밝은색은 완전히 사라지고 어두운색만을 가지게 된다. 그림 3-(a)는 기존의 Mean Shift 알고리즘을 이용한 객체추적의 예를 보여준다. 그림 3-(a)의 두 번째와 세 번째 행의 결과 영상에서 초기의 색상이 있는 곳을 중심으로 계속 비슷하게 추적함을 알 수 있다. 그러나 초기 색상이 완전히 사라진 네 번째 결과 영상에서는 올바르게 추적하지 못함을 알 수 있다. 그러므로 객체추적을 위해 초기 설정한 색상만을 사용하는 기존의 Mean Shift 알고리즘은 급격한 색상변화를 가지는 객체추적이 어렵다는 것을 이 실험을 통해 확인할 수 있다.

반면 그림 3-(b)는 제안 알고리즘을 이용하여 추적한 결과 영상이다. 제안 알고리즘은 추적 대상의 색상변화를 감지하고 변화된 색상 정보를 갱신하여 추적에 활용한다. 그래서 그림 3-(b)의 실험 결과와

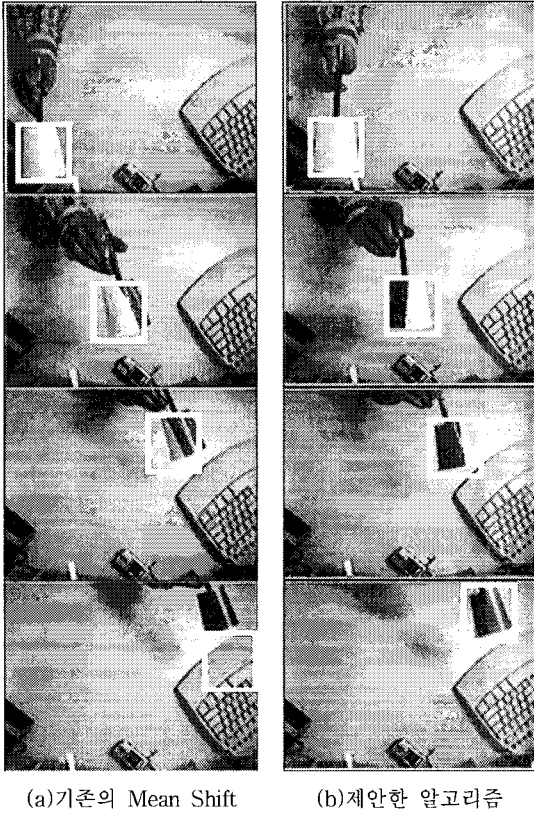


그림 3. 기존 방법과 제안한 알고리즘을 이용한 간단한 객체 추적 비교 예

같이 추적 객체의 큰 색상변화에 상관없이 정확히 추적한다. 이 실험결과를 통해 기존의 색상 기반 추적 알고리즘과 비교하여 제안 알고리즘이 색상변화에 더 강하다는 것을 알 수 있다.

보다 객관적인 성능 비교를 위하여 그림 3의 간단한 배경에서 실세계의 복잡한 배경으로 변경하여 실험한다. 추적 대상은 일상생활에서 흔히 사용하는 컵을 대상으로 하였다. 성능비교를 위하여 기존 Mean Shift 알고리즘과 제안 알고리즘을 사용하여 실험한 결과를 각각 그림 4와 그림 5에서 보여주고 있다.

그림 4는 기존 Mean Shift 알고리즘을 이용한 결과 영상이다. 그림 4-(a)부터 (f)까지는 초기에 설정한 밝은색이 일정상 남아 있기 때문에 객체의 위치를 비교적 잘 추적한다. 그러나 그림 4 (g)-(i) 경우 초기 설정한 밝은색만을 기준으로 객체를 추적하기 때문에 초기 설정한 밝은색이 반 이상 사라지는 그림 4-(g)부터는 점점 더 많은 오류를 나타낸다. 결국 초

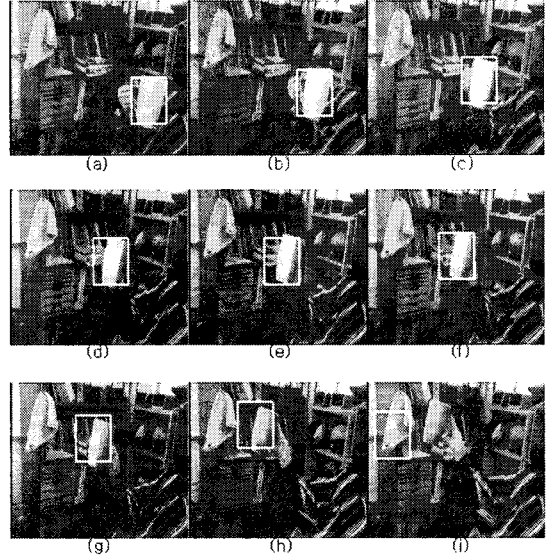


그림 4. 기존 Mean Shift 알고리즘을 이용한 컵 추적 결과 영상. (a)에서(i)까지 순차영상

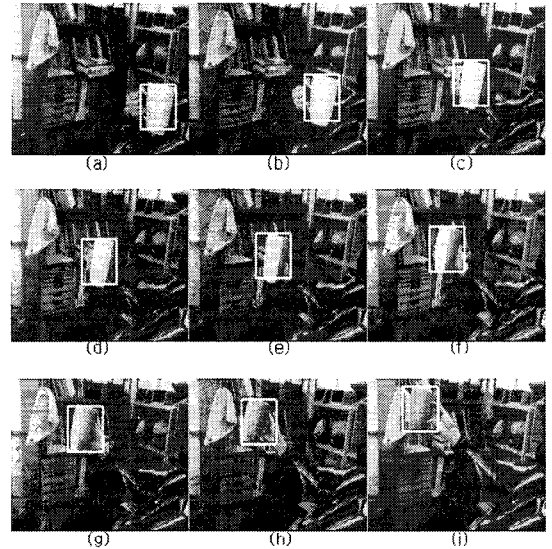


그림 5. 제안 알고리즘을 이용한 컵 추적 결과 영상. (a)에서 (i)까지 순차영상

기 설정한 밝은색이 사라진 그림 4-(i)에서는 초기 설정한 밝은색과 유사한 노란색 수건을 객체 위치로 오판하게 된다.

그림 5는 그림 4와 동일한 외부 환경에서 동일한 객체에 대해 제안 알고리즘으로 실험한 결과를 보여 준다. 그림 4에서 보여준 결과와는 달리 초기 설정한

밝은색(노란색)이 사라지기 시작하는 그림 5-(d)부터 완전히 사라지는 (i)까지 객체의 위치를 정확히 추적한다. 제안 알고리즘이 복잡한 배경에서도 그림 3에서 보여준 실험결과와 동일한 성능을 유지함을 그림 5를 통해 확인할 수 있다.

그림 6은 인공적인 객체가 아닌 사람에 대해 기존의 Mean Shift 알고리즘과 제안한 방법으로 추적한 결과이다. 대상 객체는 사람 얼굴을 대상으로 하였다. 앞선 실험과 마찬가지로 다양한 색상을 가지는 복잡한 배경에서 동일한 조건으로 두 개의 알고리즘

을 비교 실험하였다. 일반적으로 사람의 얼굴을 추적하는 경우 발생하는 문제점 중 하나는 추적 대상이 카메라의 반대 방향으로 고개를 돌리는 경우에 발생한다. 이 경우 초기에 설정한 피부색이 머리색으로 바뀌게 되는데 피부색과 얼굴색이 다르다면 기존 Mean Shift 알고리즘을 이용할 경우 정확한 추적을 하기 어렵다.

그림 6-(a)는 기존의 Mean Shift 알고리즘을 이용해 객체를 추적한 결과이다. 보는바와 같이 카메라 반대 방향으로 얼굴을 돌리는 경우 초기에 설정한 정면 얼굴의 피부색이 전혀 보이지 않고 머리색에 해당하는 검은색으로 변하게 된다. 이 경우 그림 3에서 본 결과처럼 기존 Mean Shift 알고리즘은 초기 색상을 기준으로 찾기 때문에 처음 설정한 피부색이 완전히 사라지는 마지막 영상에서는 객체를 정확히 추적하지 못하게 된다. 본 논문에서 제안한 방법으로 추적한 결과를 보여주는 그림 6의 (b)에서는 추적 대상이 피부색에서 머리색으로 완전히 색상이 변해도 성공적으로 추적을 완수한다. 그림 6을 통해 복잡한 배경을 가진 실세계의 객체를 대상으로 제안한 알고리즘을 이용할 경우 객체의 색상정보가 변하여도 정확한 추적이 가능함을 입증하였다.

그림 7은 앞선 그림 5의 실험에서 제안 알고리즘이 추적한 객체를 정의된 검색창의 크기만큼 잘라낸 영상이다. 그림 7-(4)부터 컵의 색상이 초기의 밝은색(노란색)에서 차츰 어두운색(빨간색)으로 변하는 것을 볼 수 있다. 일곱 번째 영상부터 밝은색이 절반 이상 사라지기 시작하여 마지막 그림 7-(9)에서는 완전히 사라지고 어두운색만이 남아 있음을 눈으로 확인할 수 있다.

그림 8은 두 번째 실험에서 제안 알고리즘을 이용하여 추적한 객체가 가지는 색상변화를 그래프로 보여주고 있다. 앞서 언급한 수식 (9)를 이용하여 그림 7의 첫 영상에서 나머지 영상들과의 색상변화도를



(a)기존의 Mean Shift 알고리즘 (b)제안한 알고리즘  
그림 6. 기존 방법과 제안한 알고리즘을 이용한 실세계 객체 추적의 예



(1) (2) (3) (4) (5) (6) (7) (8) (9)

그림 7. 그림 5의 실험에서 제안 알고리즘이 추적한 객체의 색상변화 영상. (1)→(9)



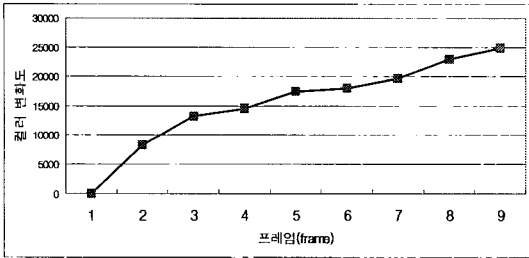


그림 8. 그림 5의 제안 알고리즘이 추적한 객체의 색상변화도

계산하여 그래프로 나타낸 것이 그림 8의 그래프이다. 그림 8의 X축은 초당 추출한 한 영상을 시간 순서에 따라 번호를 부여하였다. X축의 값은 그림 7의 (1)부터 (9)까지의 번호와 서로 대응된다. Y축은 수식 (9)를 통해 구한 색상변화도를 표시한다. 색상변화도를 구하기 위해 그림 7-(1)부터 (9)까지의 영상에 대해 RGB 3개의 히스토그램을 구하고 이를 3차원의 함수로 재구성한다. 이 3차원 함수들 중에서 초기 그림 7-(1)의 함수를 기준으로 그림 7-(1)부터 (9)까지의 함수와의 차이를 구하면 그림 8과 같은 그래프가 나오게 된다. 그림 8을 통해 객체의 색상변화가 매 영상별로 계속적으로 일정 값 이상 변하고 있는 것을 볼 수 있다. 이 그래프를 통해 처음과 마지막 영상 사이에 현격한 색상변화가 있음을 확인할 수 있다.

그림 9는 기존의 Mean Shift 알고리즘과 제안 알고리즘의 추적 성능을 비교한 그래프이다. 이 그래프의 X축은 시간에 따른 영상의 일련번호로써 앞선 그림 7과 8의 영상번호와 각각 동일하게 대응되며 Y축은 오차를 의미한다. 오차 측정은 선정된 각 영상의 물체 중심점을 정의하고 Mean Shift 알고리즘과 제

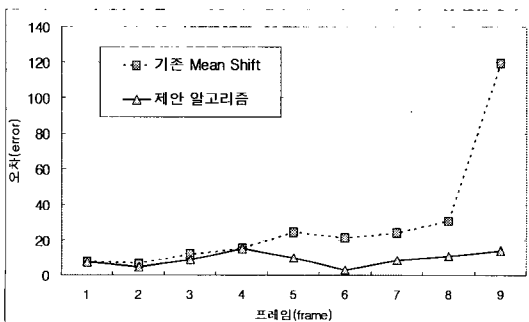


그림 9. 기존 Mean Shift 알고리즘과 제안 알고리즘간의 성능 비교

안 알고리즘이 추정한 객체의 중심 위치간의 거리를 유클리드 거리(Euclidean Distance)로 환산하여 추정하였다. 그림 7, 8의 영상 및 그래프와 병행하여 보면 본 실험에서는 색상변화의 정도가 심하지 않은 영상 1번부터 4번까지는 Mean Shift 알고리즘과 제안 알고리즘은 비슷한 추적 성능을 가짐을 알 수 있다. 초기 색상이 본격적으로 바뀌기 시작하는 영상 프레임 5번부터 두 알고리즘 간의 성능 격차가 조금씩 드러나기 시작해서 영상 프레임이 진행될수록 그 격차는 차츰 커지고 마지막 영상 프레임 9번에서는 현격한 성능 차이를 보인다. 그림 9의 성능 비교 그래프를 통해 제안 알고리즘이 급격한 색상변화에도 강함을 객관적으로 확인할 수 있다.

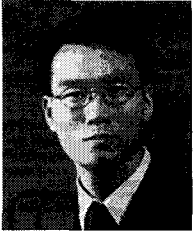
### 5. 결론

본 논문은 기존의 색상 기반 Mean Shift 알고리즘을 이용한 문제점 중 하나인 큰 색상변화를 보이는 객체추적 알고리즘을 제안하였다. 제안한 알고리즘은 Mean Shift 알고리즘을 수정하여 객체추적을 수행하면서 추적 객체의 색상변화도를 측정한다. 만약 색상변화도가 일정 값 이상일 경우 갱신한다. 추적 대상의 색상변화도를 감지하고 이를 반영함으로써 기존의 색상 기반 알고리즘이 가진 문제점을 해결하였다. 이를 입증하기 위해 본 논문에서는 간단한 배경과 복잡한 배경에서 실험객체와 실제세계의 객체를 대상으로 실험하였다. 모든 실험객체는 위치이동과 동시에 색상변화를 동시에 가진다. 동일한 객체를 대상으로 동일한 외부환경에서 기존의 Mean Shift 알고리즘과 제안 알고리즘으로 추적을 수행하였다. 기존 색상 기반 Mean Shift 추적 알고리즘은 초기 설정된 색상이 사라지는 큰 색상변화를 가진 객체에 대해 정확한 추적을 할 수 없었다. 그러나 색상변화를 감지하고 갱신하는 제안 알고리즘은 추적 객체가 급격한 색상변화를 가지더라도 정확히 추적한다는 것을 실험을 통해 입증하였다.

우리는 이 방법을 일반적인 객체추적과 추적 대상의 색상변화 정도를 알 수 없을 경우 효과적으로 사용할 수 있으리라 기대한다. 향후 추적대상의 색상정보 이외의 다른 객체가 갖는 데이터 특성을 활용하여 색상정보로 추적하기 어려운 환경에서도 강한 방법의 연구가 요구된다.

## 참 고 문 헌

- [1] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Transaction Pattern Analysis Machine Intelligence*, Vol. 25, No. 5, pp. 564-575, 2003.
- [2] I. Matthews, T. Ishikawa, and S. Baker, "The Template Update Problem," *IEEE Transaction Pattern Analysis Machine Intelligence*, Vol. 26, No. 6, pp. 810-815, 2004.
- [3] Y. Boykov and D. Huttenlocher, "Adaptive Bayesian Recognition in Tracking Rigid Objects," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 697-704, 2000.
- [4] S. McKenna, Y. Raja, and S. Gong, "Tracking Colour Objects Using Adaptive Mixture Models," *Image and Vision Computing*, Vol. 17, pp. 223-229, 1999.
- [5] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, Vol. 1, pp. 415-422, 2001.
- [6] N. Gordon and D. Salmond, "Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter," *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 6, pp. 1434-1443, 1995.
- [7] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An Adaptive Color-Based Particle Filter," *Image and Vision Computing*, Vol. 21, No. 1, pp. 99-110, 2003.
- [8] K. Fukunaga and L. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Transaction on Information Theory*, Vol. 21, No. 1, pp. 32-40, 1975.
- [9] Y. Cheng, "Mean shift, Mode Seeking, and Clustering," *IEEE Transaction Pattern Analysis Machine Intelligence*, Vol. 17, pp. 790-799, 1995.
- [10] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of Non-rigid Objects Using Mean Shift," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 142-149, 2000.
- [11] D. Comaniciu and V. Ramesh, "Mean shift and Optimal Prediction for Efficient Object Tracking," *Proceedings of International Conference on Image Processing*, pp. 70-73, 2000.
- [12] F. Porikli and O. Tuzel, "Human Body Tracking by Adaptive Background Models and Mean Shift Analysis," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, March 2003.
- [13] R.V. Babu, P. Perez, and P. Bouthemy, "Robust Tracking with Motion Estimation and Kernel-Based Color Modeling," *Proceedings of International Conference on Image Processing*, Vol. 1, pp. 717-720, Sept. 2005.
- [14] D. Comaniciu and P. Meer, "Robust Analysis of Feature Spaces: Color Image Segmentation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 750-755, 1997.
- [15] D. Comaniciu and P. Meer, "Mean Shift Analysis and Applications," *Proceedings of 7th International Conference on Computer Vision*, Greece, pp. 1197-1203, 1999.
- [16] D.W. Scott, "Multivariate Density Estimation", New York: Wiley, 1992.
- [17] D. Comaniciu, "Nonparametric Robust Methods for Computer Vision," PhD thesis, Department of Electrical and Computer Engineering, Rutgers University, 1999.
- [18] D. Comaniciu and P. Meer, "Mean Shift : A Robust Approach Toward Feature Space Analysis," *IEEE Transaction Pattern Analysis Machine Intelligence*, Vol. 24, No. 5, pp. 603-619, 2002.



최 광 남

- 1988년 중앙대학교 전자계산학과 학사
- 1990년 중앙대학교 전자계산학과 대학원 석사
- 2002년 영국, Univ. of York, Dept. of Computer Sci-

ence 박사

1990년~1995년 국방과학연구소, 연구원  
 2002년~현재 중앙대학교 컴퓨터공학부 교수  
 관심분야 : 영상처리, 컴퓨터비전, 패턴인식



황 인 택

- 2003년 중앙대학교 컴퓨터공학과 공학사
- 2005년 중앙대학교 컴퓨터공학과 대학원 석사
- 2005년~현재 중앙대학교 컴퓨터공학과 박사과정

2000년~2002년 (주)네오엑트 온라인 게임개발  
 관심 분야 : 객체추적, 영상처리, 영상 인식