

# 지역정보와 색 정보의 계층적 적용에 의한 능동 객체 추적

정준용\*, 이규원\*

\*대전대학교 정보통신공학과

e-mail: assarow@nate.com , kwlee@dju.kr

## Active Object Tracking based on hierarchical application of Region and Color Information

Joon-Yong Jeong\*, Kyu-Won Lee\*

\*Dept of Information & Communications Engineering, Daejeon University

### 요 약

본 논문에서 Pan, Tilt 카메라를 이용한 객체 추적을 위하여 초기 지역정보를 이용하여 객체를 검출하고 검출된 객체의 색 정보를 이용하여 능동 객체를 추적하는 기술을 제안한다. 외부 환경의 잡음을 제거하기 위해 적응적인 가우시안 혼합 모델링을 이용하여 배경과 객체를 분리한다. 객체가 정해지면 카메라가 이동하는 동안에도 추적이 가능한 CAMShift 추적 알고리즘을 이용하여 객체를 실시간으로 추적한다. CAMShift 추적 알고리즘은 객체의 크기를 계산하므로 객체의 크기가 변하더라도 유동적인 객체 판별이 가능하다. Pan, Tilt의 위치는 구좌표계(Spherical coordinates system)를 이용하여 계산하였다. 이렇게 구해진 Pan, Tilt 위치는 Pan, Tilt 프로토콜을 이용하여 객체의 위치를 화면의 중심에 놓이게 함으로써 적합한 추적을 가능하게 한다.

## 1. 서론

최근 국내외적으로 사회 안전망 구축에 많은 관심이 고조되고 있어, 보안·감시시장이 급속하게 성장되고 있다. 이에 따라 영상을 효과적으로 처리하여 물체를 효율적으로 감시하는 영상 감시 시스템에 대한 연구가 활기를 띠고 있다[1][2]. 감시 카메라의 종류를 3가지로 나눠 보면 고정 카메라, Pan, Tilt 카메라, 전 방위 카메라로 나눌 수 있다. 본 논문에서는 Pan, Tilt 카메라를 이용함으로써 카메라 주변의 모든 영역 및 원거리, 근거리에 대한 감시감독을 가능하도록 하여 고정 카메라가 가지는 단점을 해결할 수 있다.

배경영상에서 객체를 검출한 후 객체들 간의 겹침 현상 시 객체를 판별하지 못하거나 겹침 현상 후 객체를 놓치게 되어 새로운 객체로 오인하는 경우가 발생한다. 본 논문에서는 이러한 문제점을 해결하기 위해 색 정보 기반의 CAMShift 알고리즘을 이용하여 객체를 정확히 판별 할 수 있게 하고 이를 Pan, Tilt 카메라로 추적하는 방법을 제안한다.

## 2. 관련이론

### 2.1 객체의 지역기반 추적

배경에서 객체를 추출한 후 객체를 계속해서 추적하는 방법 중에 객체의 지역정보를 이용한 추적방법이

ADVISOR의 Reading People Tracker[3]에 사용된다. 배경에서 객체를 추출할 때 잡음으로 인해 객체가 정확하게 추출되지 못하는 부분을 해결한다. 예를 들어 비슷한 픽셀 값의 배경과 객체가 겹치게 되거나 두 개 이상의 객체가 겹치면 실제 객체의 지역 정보는 정확하지 못하게 된다. 따라서 추적된 객체의 지역정보를 정확히 판단하기 위해 두 개의 지역을 나누는 방법을 사용한다.

### 2.2 색 정보를 이용한 추적

CAMShift(Continuously Adaptive Mean Shift)[4]알고리즘의 기반이 되는 MEANShift[5] 알고리즘은 고정된 윈도우 크기를 가지고 색 정보에 기반하여 현재 프레임에서의 추적대상의 중심 위치로 수렴해 가는 방법이다. CAMShift는 기본적으로 MEANShift 알고리즘을 사용하면서 목표의 크기 및 각도를 같이 계산한다. 원하는 객체의 색(Hue) 정보를 추출하여 계속되는 영상에서 색 정보를 비교하여 원하는 부분을 추적할 수 있도록 하는 알고리즘이다.

CAMShift[6] 알고리즘은 다음과 같이 동작한다.

[단계 1] 탐색윈도우의 초기 위치를 정한다.

[단계 2] 컬러 확률 분포를 계산하고, 탐색 윈도우의 중심을 찾기 위해 MEANShift 알고리즘을 수행한다. 여기에서 객체의 크기 및 각도는 탐색 윈도우 내 컬러 확률 분포의 1차, 2차 모멘트를 계산하여 구할 수 있다. 1차 모멘

트  $M_{10}$ 와 2차 모멘트  $M_{20}$ 는 식 (1), (2)와 같이 나타낼 수 있다

$$M_{10} = \sum_x \sum_y xI(x,y), M_{01} = \sum_x \sum_y yI(x,y) \quad (1)$$

$$M_{20} = \sum_x \sum_y x^2I(x,y), M_{02} = \sum_x \sum_y y^2I(x,y) \quad (2)$$

여기서  $I(x, y)$ 는  $(x, y)$ 위치에서의 픽셀 값을 나타낸다.  
 [단계 3] 다음 프레임에서 탐색 윈도우의 위치와 크기를 [단계 2]에서 얻은 중심으로 이동한다.  
 [단계 4] MEANShift 알고리즘이 수렴할 때까지 새로운 탐색 윈도우에 대하여 [단계 2~3]과정을 반복한다.

### 3. 제안하는 시스템

#### 3.1 지역정보와 색 정보의 계층적 적용에 의한 객체 추적

2절에서 살펴본 지역 정보를 이용한 객체 추적 방법은 Pan, Tilt 카메라를 이용한 객체 추적에 적용할 경우 문제점이 발생한다. Pan, Tilt 카메라가 이동할 경우 배경영상과 객체가 하나의 물체로 오인되는 현상이 발생된다. 색 정보를 이용할 경우 Pan, Tilt 카메라가 이동을 하더라도 배경과 객체의 분리가 가능하다는 점에 착안하여 지역정보와 색 정보를 계층적으로 사용하여 정확한 객체 추적이 가능한 알고리즘을 제안한다.

본 논문에서 제안한 시스템은 기존에 제안된 배경 제거 기법을 기본으로 하여 설계한다. 제안하는 시스템은 그림 1과 같다.

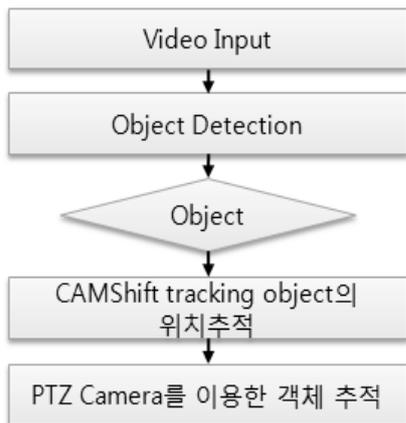


그림 1. 시스템 블록도

제시된 방법인 계층적 객체 추적 알고리즘의 기본적인 추적 방법에는 지역정보에 근거한 추적 방법을 사용한다. 입력영상 그림 2(a)로부터 객체를 검출하여 그림 2(b)와 같이 객체의 윤곽선을 검출한다.



그림 2. (a) 입력영상 (b) 윤곽선 검출

그리고 검출된 윤곽선을 이용하여 초기 탐색윈도우의 위치를 객체의 중심에 가깝게 설정 한다. 만일 탐색윈도우 내 컬러 확률분포 중 객체 주위 배경의 컬러 확률분포도가 높게 되면, 객체가 아닌 배경을 추적하기 때문이다. 그림 3과 같이 윤곽선 위 점의 위치를 계산하여 객체의 중심에 가깝게 탐색윈도우를 설정한다.

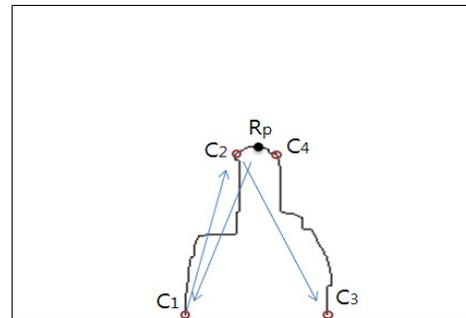


그림 3. 윤곽선 위 좌표 계산

윤곽선 위에  $n$ 개의 점이 있을 때 윤곽선 위 임의의 점  $R_p(x_p, y_p)$ 로부터 가장 멀리 있는 점  $C_1(x_{c1}, y_{c1})$ 을 구한다. 점  $C_1(x_{c1}, y_{c1})$ 은 점  $R_p(x_p, y_p)$ 로부터 식(3)을 만족하는  $n$ 번째 점이다.

$$C_n = (x_n, y_n)$$

$$C_{\max} = \max \left[ \sqrt{((x_p - x_n)^2 + (y_p - y_n)^2)} \right] \quad (3)$$

같은 방법으로 윤곽선 위 점들 중 점  $C_1(x_{c1}, y_{c1})$ 로부터 가장 멀리 있는 점  $C_2(x_{c2}, y_{c2})$ 를 구할 수 있다. 세 번째 점  $C_3(x_{c3}, y_{c3})$  또한 점  $C_1(x_{c1}, y_{c1})$ ,  $C_2(x_{c2}, y_{c2})$ 에서 가장 먼 점을 구한다. 이렇게 구해진 좌표값을 이용하여 객체에 중심에 가까운 새로운 탐색윈도우의 좌표를 계산할 수 있다. 점  $C_2(x_{c2}, y_{c2})$ 에서 점  $C_4(x_{c4}, y_{c4})$ 가 객체의 머리영역의 가로 길이라고 가정할 때, 탐색윈도우의 나머지 점  $(x_{c2}, y_{c4})$ ,  $(x_{c4}, y_{c3})$ 를 얻을 수 있다. 새롭게 설정된 탐색윈도우를 이용하여 그림 4(b)와 같이 객체의 중심에서 색(Hue)히스토그램을 얻을 수 있다.

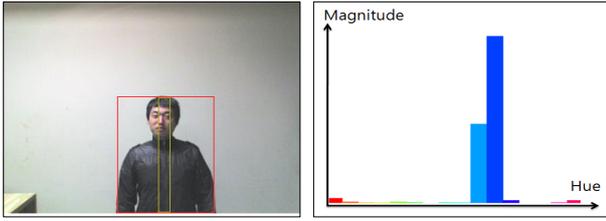


그림 4. (a)탐색윈도우 설정  
(b)추적된 객체의 색(Hue)히스토그램

얻어진 색(Hue)정보 히스토그램을 CAMShift 알고리즘에 적용을 하고 그림 5와 같이 색(Hue)정보를 이용한 추적을 시작한다.



그림 5. CAMShift 알고리즘을 이용한 추적

### 3.2 Pan, Tilt 위치 제어

카메라의 좌표는 그림 6과 같이 X축, Y축, Z축으로 표시되고 카메라는 -Z축을 향한다고 가정한다.

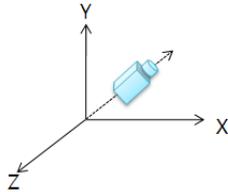


그림 6. 카메라 좌표

그림 7에서 화면상의 이동해야 하는 위치가  $(u_p, v_p)$  일 경우 전체 화면에 대한 회전각을 알고 있으면  $(u_p, v_p)$ 에서 식(4)처럼  $\alpha, \beta$  를 구할 수 있다.

$$\alpha = \frac{\theta_w}{W} \cdot u_p, \beta = \frac{\phi_H}{H} \cdot v_p \quad (4)$$

처음 초기 좌표 -Z축을  $(0,0,-1)$ 로 설정한 후에 초기 좌표를 카메라 좌표로 변환하면 그림 7에 나타난 바와 같이 되고, 식(5)와 같이 표현할 수 있다.  $R(Y_w, \alpha)$ 는 Y축을 기준으로  $\alpha$ 만큼의 회전을 의미하고  $R(X_w, \beta)$ 는 X축을 기준으로  $\beta$ 만큼의 회전을 의미한다.

$$\begin{aligned} P_c &= [00-11] \cdot R(Y_w, \alpha) \cdot R(X_w, \beta) \\ &= [00-11] \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) \\ &= [-\sin\alpha \cos\beta \quad \sin\alpha \sin\beta \quad -\cos\alpha \cos\beta \quad 1] \end{aligned}$$

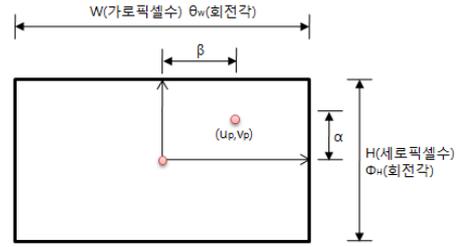


그림 7. 입력영상과 카메라 이동의 관계

여기서  $P_w$ 는 초기 좌표  $(0,0,-1)$ 을 말하며,  $P_c$ 는 카메라 좌표를 말한다. 카메라 좌표를 전체 좌표로 변환하면 식(6), 식(7)과 같이 표현된다.  $R(x_c, \phi_c)$ 는 X축을 기준으로  $\phi_c$ 만큼 회전을 의미하고  $R(y_c, \theta_c)$ 는 Y축을 기준으로  $\theta_c$ 만큼 회전을 의미한다.

$$\begin{aligned} P_{w,\phi} &= R(x_c, \phi_c) \cdot P_c \\ P_{w,\theta} &= R(y_c, \theta_c) \cdot P_c \end{aligned} \quad (6)$$

$$\begin{aligned} P_w &= P_c \cdot P_{w,\theta} \cdot T_{w,\phi} \\ &= P_c \cdot \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta_c & 0 & \cos\theta_c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi_c & \sin\phi_c & 0 \\ 0 & -\sin\phi_c & \cos\phi_c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7) \end{aligned}$$

여기서  $\theta_c$ 는 Pan의 각도이며,  $\phi_c$ 는 Tilt의 각도이다. 식(7)을 이용하여 이동해야 할 최종 위치  $P_w(X_w, Y_w, Z_w)$  값을 구하면 최종 Pan( $\theta$ ), Tilt( $\phi$ ) 위치는 식(8)과 같이 된다.

$$\begin{aligned} \theta &= \tan^{-1}(X_w/Z_w) \\ \phi &= \sin^{-1}(Y_w) \end{aligned} \quad (8)$$

Pan, Tilt 카메라를 이용한 객체 추적을 위해선 정확한 객체의 위치를 찾고 객체의 위치를 화면의 중앙으로 위치시켜야 한다. 추적하는 객체를 화면의 중앙으로 이동시키기 위해선 Pan, Tilt의 위치를 제어함으로써 가능하다.

### 4. 실험 및 결과 고찰

임의의 객체가 이동하는 환경에서 LAN 인터페이스의 Pan, Tilt 카메라를 이용하여 실시간으로 단일 객체를 능동 추적하는 실험을 행하였다. 사용된 이미지의 해상도는  $320 \times 240$ 이다. 그리고 기존의 탐색윈도우의 재설정 없이 CAMShift 알고리즘과 논문에서 제안한 방식을 비교하였다. 총 3000프레임간 능동객체를 대상으로 비교하였다. 프레임 재생 속도는 평균 30(frame/sec)이다.

기존 방식의 경우 그림 8과 같이 배경의 컬러 확률분포가 높아 객체가 아닌 배경을 추적하는 것을 볼 수 있다.



그림 8. 기존 CAMShift 알고리즘을 이용한 추적

하지만 제안하는 알고리즘을 사용한 그림 9는 객체의 움직임에 따라 탐색 윈도우의 크기도 변하면서 잘 추적되고 있음을 알 수 있으며, 초기 추적을 하던 객체 주변에 다른 객체가 나타나 겹침 현상이 나타났을 때 겹침 후에도 정확히 객체를 계속해서 추적하는 것을 볼 수 있다. 그리고 Pan, Tilt 카메라가 이동하는 동안에도 초기 객체를 놓치지 않고 계속해서 정확한 추적을 할 수 있는 것을 알 수 있다.

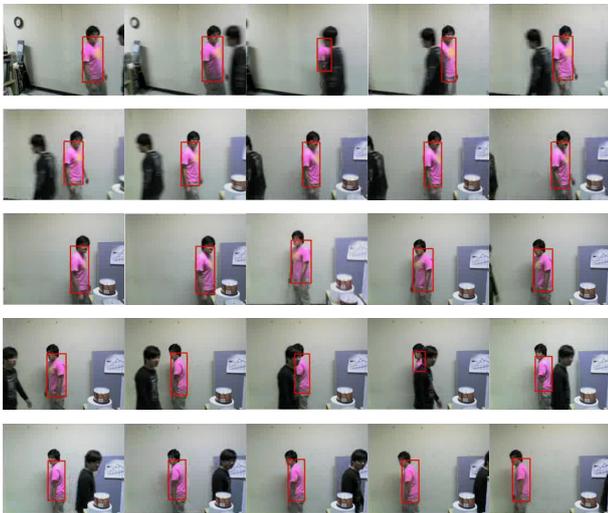


그림 9. 계층적 적용을 통한 추적

Algorithm	기존 CAMShift만 적용한 알고리즘	제안하는 알고리즘
정확도	68.45	91.36

표 1. 각 알고리즘에 추적 정확도 비교(%)

표 1은 실제 추적한 프레임 수를 비교하여 정확도를 나타낸 것이다. 식 (9)로 구하였다.

$$Accuracy(\%) = \frac{T_f}{RT_f} \times 100 \quad (9)$$

여기에서  $RT_f$ 는 실제로 객체를 추적해야 하는 프레임 수를 의미하고,  $T_f$ 는 실험을 통해서 객체가 추적된 프레임 수를 뜻한다. 기존 CAMShift 알고리즘만을 적용한 객체 추적은 대략 70%의 정확도를 보였다. 이는 탐색윈도우가 초기 객체가 아닌 배경

을 추적하거나 한번 놓치면 다시 추적하기가 힘들다는 것을 의미한다. 이에 반해 제안한 방법을 사용한 경우에는 90%이상의 정확도를 보이는 것을 확인하였다. 또한 탐색 윈도우의 위치 및 크기를 검출된 객체의 정보를 가지고 자동 할당함으로써 초기에 지정해야하는 번거로움을 없앴다.

### 5. 결론

제안하는 방법은 초기 지역 정보를 이용하여 객체를 검출하고, 검출된 객체의 중심에 가깝게 탐색 윈도우를 설정한다. 새롭게 설정된 탐색윈도우를 통해 객체의 색 확률 분포를 측정하고, 이를 CAMShift 알고리즘에 적용한다. 화면상 객체의 위치를 찾는 다음 구좌표계(Spherical coordinates system)를 이용하여 Pan, Tilt의 위치를 구한다. 구해진 Pan, Tilt의 위치를 제어하여 객체를 추적하는 능동추적 시스템을 제안하였다.

그리고 실험을 통하여 기존 탐색윈도우의 재설정 없는 CAMShift 알고리즘과 비교하였을 때 제안하는 알고리즘은 약 20%의 성능 향상을 볼 수 있었다. 그러나 색 정보만을 이용하기 때문에 유사한 색상에 대해 오류율이 높은 단점이 존재한다. 실제 화면상의 객체 위치를 제어하기 때문에 객체가 움직일 경우 중앙에 위치하기 위한 객체 예측이 필요하다. 추후 연구에는 위의 문제점을 보완한 강건한 추적을 수행할 수 있는 알고리즘의 연구를 계속 진행할 예정이다.

### 참고문헌

[1] Gian Luca Foresti, Christian Micheloni, Lauro Sindaro, Paolo Remagnino, and Tim ellis "Active Video-Based Surveillance System" IEEE Signal Processing Magazine, pp. 25-37, March 2005.

[2] Arun Hampapur, Lisa Brown, Jonathan Connell, Ahmet Ekin, Norman Hass, Max Lu, Hans Merkl, Sharath Pankanti, Andrew Senior, Chiao-Fe Shu, and Ying Li Tian "Smart Video Surveillance Applications", IEEE Signal Processing Magazine, pp. 38-51, March 2005.

[3] Nils T Siebel, "Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications", The University of Reading 2003.

[4] J. G. Allen, R.YD Xu, J. S. Jin "Object Tracking using CamShift Algorithm and Multiple Quantized Feature Spaces," Pan-Sydney Area Workshop on Visual Information Processing, pp. 3-7, 2003

[5] D Comaniciu, V Ramesh, P Meer "Real-time tracking of non-rigid objects using mean shift" Computer Vision and Pattern Recognition, 2000. Processings. IEEE Conference Vol.2 pp.142-149. 2000

[6] Nicole M. Artner, "A Comparison of Mean Shift Tracking Methods", CESCg 2008, pp. 197-204, 2008