

# 중심 이동 기반의 스케일 적응적 물체 추적 알고리즘

이석호<sup>†</sup>, 최은철<sup>\*\*</sup>, 강문기<sup>\*\*\*</sup>

## 요 약

본 논문에서는 물체를 구성하고 있는 컬러들의 기하학적 중심을 이용하여 물체의 크기에 적응적인 추적 알고리즘을 제안한다. 대부분의 크기 적응적 알고리즘은 표적윈도우의 크기를 결정하기 위하여 히스토그램을 사용한다. 그러나, 이러한 방법은 표적의 배경에 표적의 색상과 유사한 물체가 존재하거나 표적의 일부분이 폐색되었을 때 표적의 크기를 추정하는데 실패한다. 이것은 히스토그램이 영역에 대한 기하학적인 공간정보를 상실한채 표적 컬러의 화소수하고만 연관되기 때문이다. 이러한 분석을 바탕으로 본 논문은 표적 컬러의 화소수의 변화에 상대적으로 덜 민감한 표적의 컬러 중심을 이용한 크기 적응 알고리즘을 제안한다. 컬러의 중심들은 공간정보를 가지고 있기 때문에 컬러중심과 표적 영역의 크기에는 직접적인 상관관계가 존재한다. 표적의 크기 변화를 추정하기 위하여 각각의 표적 컬러에 대한 줌팩터를 추정한 후, 적절한 필터링 과정을 통해 하나의 줌팩터를 추정한다. 제안한 크기 추정 알고리즘은 중심이동 기반의 추적 알고리즘과 결합된다. 제안된 크기 적응적 추적 알고리즘은 배경에 유사한 컬러가 존재하는 경우에도 안정적으로 작동하는 것을 실험으로 검증한다.

## Object Tracking Based on Centroids Shifting with Scale Adaptation

Suk-Ho Lee<sup>†</sup>, Euncheol Choi<sup>\*\*</sup>, Moon Gi Kang<sup>\*\*\*</sup>

## ABSTRACT

In this paper, we propose a stable scale adaptive tracking method that uses centroids of the target colors. Most scale adaptive tracking methods have utilized histograms to determine target window sizes. However, in certain cases, histograms fail to provide good estimates of target sizes, for example, in the case of occlusion or the appearance of colors in the background that are similar to the target colors. This is due to the fact that histograms are related to the numbers of pixels that correspond to the target colors. Therefore, we propose the use of centroids that correspond to the target colors in the scale adaptation algorithm, since centroids are less sensitive to changes in the number of pixels that correspond to the target colors. Due to the spatial information inherent in centroids, a direct relationship can be established between centroids and the scale of target regions. Generally, after the zooming factors that correspond to all the target colors are calculated, the unreliable zooming factors are filtered out to produce a reliable zooming factor that determines the new scale of the target. Combined with the centroid based tracking algorithm, the proposed scale adaptation method results in a stable scale adaptive tracking algorithm. It tracks objects in a stable way, even when the background colors are similar to the colors of the object.

**Key words:** Object Tracking(컬러중심), Color Centroid(추적), Scale Adaptation(스케일 적응)

※ 교신저자(Corresponding Author): 강문기, 주소: 서울 특별시 서대문구 신촌동 134 연세대학교 제2공학관 727 (120-749), 전화: (02)2123-7711, FAX: (02)312-4584, E-mail: mkang@yonsei.ac.kr

접수일: 2010년 10월 28일, 수정일: 2010년 1월 5일

완료일: 2011년 3월 14일

<sup>†</sup> 정회원, 동서대학교 컴퓨터정보공학부 (E-mail: petrasuk@gmail.com)

<sup>\*\*</sup> 정회원, 연세대학교 TMS 정보기술 사업단 (E-mail: mkang@yonsei.ac.kr)

<sup>\*\*\*</sup> 정회원, 연세대학교 TMS 정보기술 사업단

※ 본 연구는 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(No.2010-0006567) 및 지식경제부, 문화체육관광부 및 정보통신연구진흥원의 IT산업원천기술개발사업의 일환으로 [2008-F-031-01, 영상 및 비디오 콘텐츠를 위한 계산사진학 기술 개발] 수행되었음.

## 1. 서 론

모바일 카메라와 지능적 감시 시스템 등의 발달로, 비정적 환경에서의 물체 추적은 최근까지 각광받는 연구분야 중 하나가 되고 있다. 비정적 카메라에 의해 취득되는 영상에서는 배경의 변화가 심하다. 따라서 정적 카메라를 사용한 추적 알고리즘에서 많이 쓰는 배경을 모델링하는 방법을 이용하여 추적하기 보다는 컬러나 텍스처 등을 이용한 추적 방법을 주로 쓰이게 된다. 일반적으로 컬러기반의 추적 방법은 초기 프레임의 표적 영역과 가장 유사한 컬러 분포를 갖는 영역을 찾음으로써, 표적의 위치를 추적한다. 고정된 크기, 혹은 적응적 크기의 표적 영역을 찾기 위해 템플릿 매칭 방법은 주로 맹목적탐색(brute-force search)을 수행하여 가장 유사도가 높은 영역을 찾는다[1-3]. 이런 템플릿 매칭방법의 계산량을 줄이기 위해 컬러의 분포와 커널 가중치를 이용하여 바타차야(Bhattacharyya)계수가 가장 높은 지점을 향해 평균이동(mean shift)하는 평균이동 기반의 추적 방법이 각광을 받게 되었다[4-6].

컬러 분포의 유사도, 특히 히스토그램의 유사도나 밝기값의 상호상관(cross-correlation)을 이용한 연구들과 달리, [7]에서는 유사도 추정 없이 컬러들의 기하학적 중심을 가중평균하여 직접 표적의 위치를 추적하는 알고리즘이 제안되었다. 컬러 중심들을 이용함으로써, 고정된 크기의 표적영역에 대한 추적이 안정적으로 수행된다. 이는 기하학적 중심들이 컬러의 공간 정보를 포함하고 있고, 화소수의 변화에 덜 민감하기 때문이다.

MeanShift 기반의 추적 방법에서 또 중요하게 다루고 있는 문제는 추적대상의 크기 추정에 관한 문제이다. 표적의 추적이 지속적으로 가능하게 하기 위해서는 영역의 크기도 적절히 추정되어야 한다. MeanShift 기반의 크기 적응 추적 방법 중 가장 잘 알려진 것은 CAMShift방법[8]이다. 이것은 표적 영역의 정규화된 히스토그램을 기반으로 표적의 크기를 추정한다. 그러나 배경에 표적과 유사한 컬러가 있는 경우 표적의 위치와 크기를 추적하는데 실패하며 배경 영역으로 퍼져 나가게 된다. [9]에서 언급된 크기 추정 방법은 여러 스케일 중에서 바타차야 계수가 가장 큰 것을 찾아 결정하는 방식이다. 그러나 이 방법은 표적 보다 작은 영역에서 표적 전체의 컬러분

포보다 더 유사한 분포를 가질 수 있으므로 표적원도가 축소되는 문제가 발생할 수 있다. 이러한 문제를 해결하기 위하여 [4]에서는 스케일 선택이론을 이용하였으나 앞에서 언급한 문제를 완전히 해결하지는 못했다. [9]에서는 표적의 방향(orientation)과 크기(scale)을 함께 고려하였지만 레벨셋(level set)함수를 이용하기 때문에 많은 계산량을 필요로 한다.

본 논문에서는 [7]에서 제안된 알고리즘과 결합된 형태의 크기 적응적 추적 알고리즘을 제안한다. 제안한 방법 또한 컬러의 중심을 이용하는 방법으로 이러한 방법은 몇 가지 장점을 지니고 있다. 우선 컬러들의 기하학적 중심은 표적의 크기와 직접적인 관계가 있기 때문에 추정 과정이 간단하고 빠르다. 또한 수많은 중심들을 이용하기 때문에 이들 중 신뢰할 수 있는 중심들만 필터링하여 사용할 수 있다. 제안된 알고리즘은 배경이 복잡하거나 일부 폐색이 발생한 상황에서도 표적의 위치 및 크기 추적을 비교적 안정적으로 수행한다. 또한 컬러들의 중심을 계산하는 과정은 많은 연산량이 요구되지 않기 때문에 실행속도도 빠르다. 2장에서는 제안한 중심기반의 크기 적응적 추적 알고리즘에 대해서 설명한다. 그리고 3장에서는 제안한 알고리즘을 이용한 실험 결과를 제시하여 본 알고리즘의 성능을 검증하고자 한다.

## 2. 크기 적응적 중심기반 추적 알고리즘

본 장에서는 우선 [7]에서 제안된 중심이동(centroid shifting) 기반의 추적알고리즘을 간단히 소개한다. 그 후, 물체의 기하학적 중심을 이용하여 표적의 크기를 추정하는 방법을 제안한다. 마지막으로 두 알고리즘을 결합하는 방법에 대해 설명한 후에 전체적인 알고리즘이 수행되는 과정에 대해 설명하도록 한다.

### 2.1 중심 이동 기반 객체 추적 알고리즘

[6]에서는 표적 컬러들의 기하학적 중심이동을 이용하여 다음 프레임의 표적 위치를 추적하는 안정적인 알고리즘을 제안하였다. 표적의 위치는 각 컬러 중심들의 가중평균으로 구해진다. 제안하는 크기 추정방법은 표적의 중심이 얼마나 정확하게 추정되는냐에 따라 성능이 좌우되기 때문에 표적의 위치를 정확하게 추정하는 것은 중요하다. 여기서 표적의 위치를 추정하는 알고리즘에 대해 간단하게 설명하고

자 한다.  $m$ 개의 컬러빈(color bin)을 갖는 물체의 중심 위치  $\hat{y}$ 는 다음과 같은 거리함수를  $\hat{y}$ 에 대해 최소화함으로써 구한다.

$$E(\hat{y}) = \sum_{u=1}^m q_u \|\hat{y} - C_u^n\|^2 \quad (1)$$

여기서  $q_u, p_u^n$ 는 각각 초기 프레임 및 현재 프레임에서 컬러빈  $u$ 의 히스토그램을 의미하며,  $n$ 은 현재 프레임을 나타내는 기호이다. 여기서  $C_u^n$ 은  $n$ 번째 프레임에서의 컬러  $u$ 에 해당하는 화소들의 위치벡터들의 평균벡터이다. 식 (1)의 거리함수  $E(\hat{y})$ 의 최소값은  $E(\hat{y})$ 을  $\hat{y}$ 에 대하여 미분한 함수를 0으로 놓음으로써, 즉  $\nabla E(\hat{y}) = \sum_{u=1}^m q_u (\hat{y} - C_u^n) = 0$ 으로 놓음으로써 구할 수 있다:

$$\hat{y} = \frac{\sum_{u=1}^m q_u C_u^n}{\sum_{u=1}^m q_u} \quad (2)$$

초기프레임에서 표적의 중심위치를 계산한 후 다음 프레임(현재 프레임이  $n$ 번째 프레임이면  $n+1$  번째 프레임)에서의 표적 위치를 찾기 위해 우선 이전에 구한 중심위치를 중심으로 검색윈도우를 설정한다.

그 후 검색영역 안에서 식 (2)를 이용하여 각 컬러들의 중심을 구하고, 초기 프레임에서 계산한 가중치를 이용하여 이들 컬러 중심들의 가중평균을 구한다. 초기 프레임에서 설정한 가중치를 사용하였기 때문에, 현재의 검색 윈도우 영역에 존재하는 컬러라 하여도, 원래 표적에 존재하지 않았다면 표적 중심위치 계산시 이 컬러에 대한 고려는 하지 않게 된다. 그러므로 현재 프레임의 컬러의 변화나 화소수의 변화에 대해 어느 정도 안정적이며, 부분적인 폐색(occlusion)에 대해서도 안정적으로 표적의 중심이 구해지게 된다. 부분적인 폐색에 대해서도 추적대상에 대한 추적이 가능한 이유는 추적대상의 컬러중에서 폐색이 되지 않은 컬러들이 비록 소수일지라도 위치 추정시에 초기 프레임에서 이 컬러들이 차지하였던 면적에 비례하여 가중치를 주기 때문에 이들의 추정에 대한 영향력이 현재 프레임에서의 화소수에 의해서가 결정되는 것이 아니기 때문이다.

## 2.2 중심의 차 벡터를 이용한 표적의 크기 추정

여기에서는 앞에서 기술한 표적의 위치 추적 알고리즘과 결합되는 크기 추정 알고리즘을 제안한다.

일단 표적의 위치가 앞 절의 알고리즘을 이용하여 계산되면 표적영역은 표적의 새로운 크기에 적응하기 위하여 크기를 변경해야한다. 대부분의 스케일 적응 알고리즘은 표적 컬러의 히스토그램을 이용하지만, 여기서는 컬러들의 기하학적 중심을 이용한다. 이는 다음의 사실들에 기인한 것이다. 우선 기하학적 중심들은 표적 컬러의 공간적 정보를 갖고 있다. 따라서 중심들은 표적 영역의 스케일과 직접적인 관련이 있다. 또한 유사한 배경 컬러로 인해 실제의 위치가 원하는 위치보다 많이 벗어난 컬러 중심들은 스케일 추정에 사용되지 않도록 쉽게 검출 및 제거가 가능하다.

스케일 추정을 위해 그림 1과 같이 표적영역을 네 개의 작은 영역으로 나눈다. 그림 1의 윗 행과 아래 행은 각각 현재 프레임과 다음 프레임을 나타내고 두 프레임 사이에는 스케일의 변화가 있다고 가정한다. 표적영역을 작은 영역으로 분할한 후 우선 각 영역에 대해 컬러들의 중심들을 각각 구한다. 그림 1(a), 1(b)는 현재 프레임에서 계산된 컬러 중심들을 나타낸 것이고, 그림 1(c), 1(d)는 다음 프레임에서

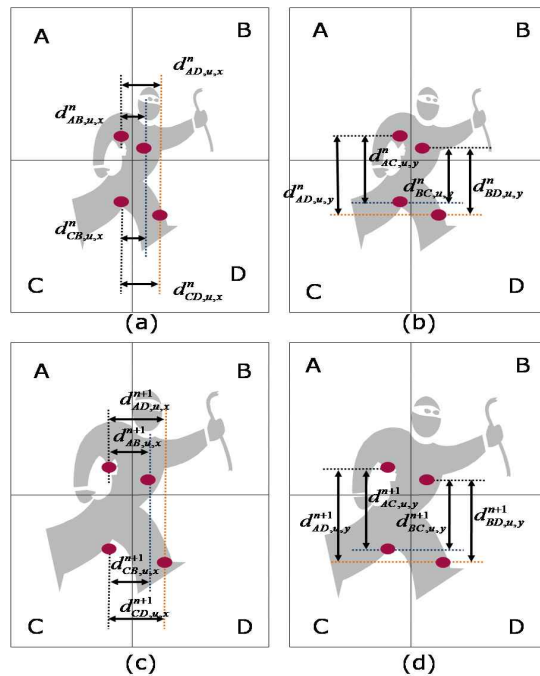


그림 1. 현재프레임과 다음 프레임의 컬러 중심간의 거리 (a) 이전 프레임의 x축으로의 스케일 추정, (b) 이전 프레임의 y축으로의 스케일 추정, (c) 현재 프레임의 x축으로의 스케일 추정, (d) 현재 프레임의 y축으로의 스케일 추정

계산된 컬러 중심들을 나타낸 것이다. 그림 1에서 볼 수 있듯 서로 다른 영역에 존재하는 컬러 중심들 간 거리는 표적 영역의 크기와 직접적인 관련이 있다. 예를 들어 그림 1(a)와 1(c)에서  $d_{AD,u,x}^n$  과  $d_{AD,u,x}^{n+1}$  의 비율은 현재 프레임과 다음프레임에 있는 표적의 폭 (width) 비율을 계산할 때 사용될 수 있다. 하나의 컬러가 아닌 여러 컬러들로부터 구한 중심들을 이용하면 보다 안정적인 리사이징 비율을 얻을 수 있다. 알고리즘의 설명을 위해  $d_{AB,u,x}^{n+1}$  을 부분 영역 A와 B 에 존재하는 컬러  $u$ 에 해당하는 중심간 거리의  $x$  성분이라고 정의한다. 즉,

$$d_{AB,u,x}^{n+1} = |C_{A,u,x}^{n+1} - C_{B,u,x}^{n+1}| \quad (3)$$

로 정의한다. 여기서  $C_{A,u,x}^{n+1}$  와  $C_{B,u,x}^{n+1}$  는 각각 다음 프레임( $n+1$ 번째 프레임)부분영역 A, B에서 컬러  $u$ 에 해당하는 화소들의 기하학적 중심의  $x$ 좌표를 의미한다. 마찬가지로

$$\begin{aligned} d_{AD,u,x}^{n+1} &= |C_{A,u,x}^{n+1} - C_{D,u,x}^{n+1}| \\ d_{CB,u,x}^{n+1} &= |C_{C,u,x}^{n+1} - C_{B,u,x}^{n+1}| \\ d_{CD,u,x}^{n+1} &= |C_{C,u,x}^{n+1} - C_{D,u,x}^{n+1}| \end{aligned} \quad (4)$$

도 정의할 수 있다. 다음으로  $x$ 축으로의 크기비율  $d_{u,x}^{n+1}$ 을 얻기 위해  $d_{AB,u,x}^{n+1}$ ,  $d_{AD,u,x}^{n+1}$ ,  $d_{CB,u,x}^{n+1}$ ,  $d_{CD,u,x}^{n+1}$  의 가중 평균을 구한다. 즉,

$$\begin{aligned} d_{u,x}^{n+1} &= W_{AB,u}d_{AB,u,x}^{n+1} + W_{AD,u}d_{AD,u,x}^{n+1} \\ &\quad + W_{CB,u}d_{CB,u,x}^{n+1} + W_{CD,u}d_{CD,u,x}^{n+1} \end{aligned} \quad (5)$$

로 구하며, 이때 가중치는 각 부분영역에 존재하는 컬러  $u$ 의 개수를 고려하여

$$W_{AB,u} = \frac{N_{A,u}N_{B,u}}{N_{A,u}N_{B,u} + N_{A,u}N_{D,u} + N_{C,u}N_{B,u} + N_{C,u}N_{D,u}} \quad (6)$$

의 형태로 구해지며,  $W_{AD,u}$ ,  $W_{CB,u}$ ,  $W_{CD,u}$ 도 동일한 방법으로 구해진다. 여기서  $N_{A,u}$ ,  $N_{B,u}$ ,  $N_{C,u}$  그리고  $N_{D,u}$ 는 각각 각 부분영역에서 컬러  $u$ 에 해당하는 화소의 개수를 의미한다.

최종적으로 컬러  $u$ 에 해당하는  $x$ 축으로의 줌팩터 (zoom factor)는

$$z_{u,x}^{n+1} = \frac{d_{u,x}^{n+1}}{d_{u,x}^n} \quad (7)$$

으로 정의된다. 여기서  $d_{u,x}^n$ 는 현재 프레임에서

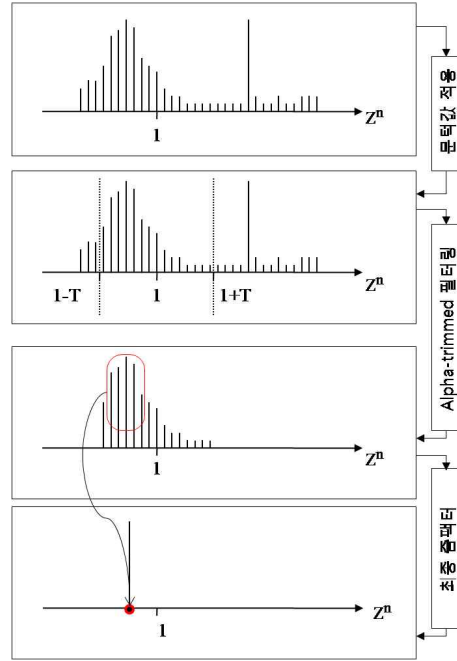


그림 2. 최종 줌팩터를 구하기 위한 과정

$d_{AB,u,x}^n$ ,  $d_{AD,u,x}^n$ ,  $d_{CB,u,x}^n$ ,  $d_{CD,u,x}^n$  을 가중 평균한 값이고,  $d_{u,x}^{n+1}$ 도 같은 방식으로 다음 프레임을 사용하여 구한다. 줌팩터  $z_{u,x}^{n+1}$ 은 컬러  $u$ 에 대한 표적의 폭( $x$ 축 방향)의 스케일 비율을 추정한 것이다.  $y$ 축에 해당하는 줌팩터도 동일한 방법으로

$$z_{u,y}^{n+1} = \frac{d_{u,y}^{n+1}}{d_{u,y}^n} \quad (8)$$

와 같이 구할 수 있다. 이렇게 구한 줌팩터  $z_{u,x}^{n+1}$ 와  $z_{u,y}^{n+1}$ 는 각각 표적의 폭과 높이의 리사이즈 비율로 사용할 수 있다. 그러나 어떤 컬러의 줌팩터는 신뢰할 수 없는 값을 갖기도 한다. 배경에 유사한 컬러가 나타남으로 특정 컬러의 줌팩터는 우리가 원하는 값보다 커지거나 작아질 수 있기 때문이다. 따라서 본 논문에서는 표적의 모든 컬러에 대해 줌팩터를 구한 후에 적절한 필터링 과정을 통해 신뢰할 수 있는 줌팩터를 선별, 이용하여 한쌍의 최종 줌팩터를 결정한다. 즉, 최종적인 줌팩터는

$$z_x^{n+1} = f(z_{1,x}^{n+1}, z_{2,x}^{n+1}, \dots, z_{U,x}^{n+1}) \quad (9)$$

와

$$z_y^{n+1} = f(z_{1,y}^{n+1}, z_{2,y}^{n+1}, \dots, z_{U,y}^{n+1}) \quad (10)$$

으로 나타낼 수 있다. 여기서 아래 첨자  $1, 2, \dots, U$ 는 각각의 컬러빈을 나타내며  $f(\cdot)$ 은 신뢰할 수 있는 줌팩터들을 필터링하는 함수이다.

함수  $f(\cdot)$ 는 여러 줌팩터 중에 1보다 너무 크거나 작은 값을 제거하는 과정을 담고 있다. 왜냐하면 연속된 프레임간에는 표적의 크기가 갑자기 변화하지 않기 때문에 이러한 줌팩터는 배경 컬러의 등장이나 표적의 폐색에 의해 발생했을 가능성이 크기 때문이다. 이렇게 선별된 줌팩터들에 대해서는 잡음제거에서 많이 사용되는  $\alpha$ -trimmed 필터링을 사용하여 최종적인 줌팩터를 구한다. 각각의 과정은 다음과 같다.

① 문턱값 적용 과정 : 1보다 너무 작거나 큰 줌팩터를 제거하는 과정이다. 결국 다음에 해당하는 줌팩

터만을 사용하게 된다.

$$\{z_{u,x}^n | 1 - T < z_{u,x}^n < 1 + T, u = 1, \dots, U\} \quad (11)$$

$$\{z_{u,y}^n | 1 - T < z_{u,y}^n < 1 + T, u = 1, \dots, U\}$$

여기서  $0 < T < 1$ 이다.

②  $\alpha$ -trimmed 과정 : ①의 과정을 통해 남은 줌팩터들에 대해  $\alpha$ -trimmed 평균[10]을 이용하여 최종적인 줌팩터를 구한다. 이 과정은 남은 줌팩터들을 정렬하여 극단에 존재하는  $\alpha$ 개의 줌팩터들을 제외한 줌팩터들에 대해 평균을 구하는 것이다. 줌팩터들의 필터링 과정을 그림 2에 나타내었다.

### 2.3 스케일에 적응적인 추적 알고리즘

중심이동 기반의 위치 추적 알고리즘과 앞 절에서

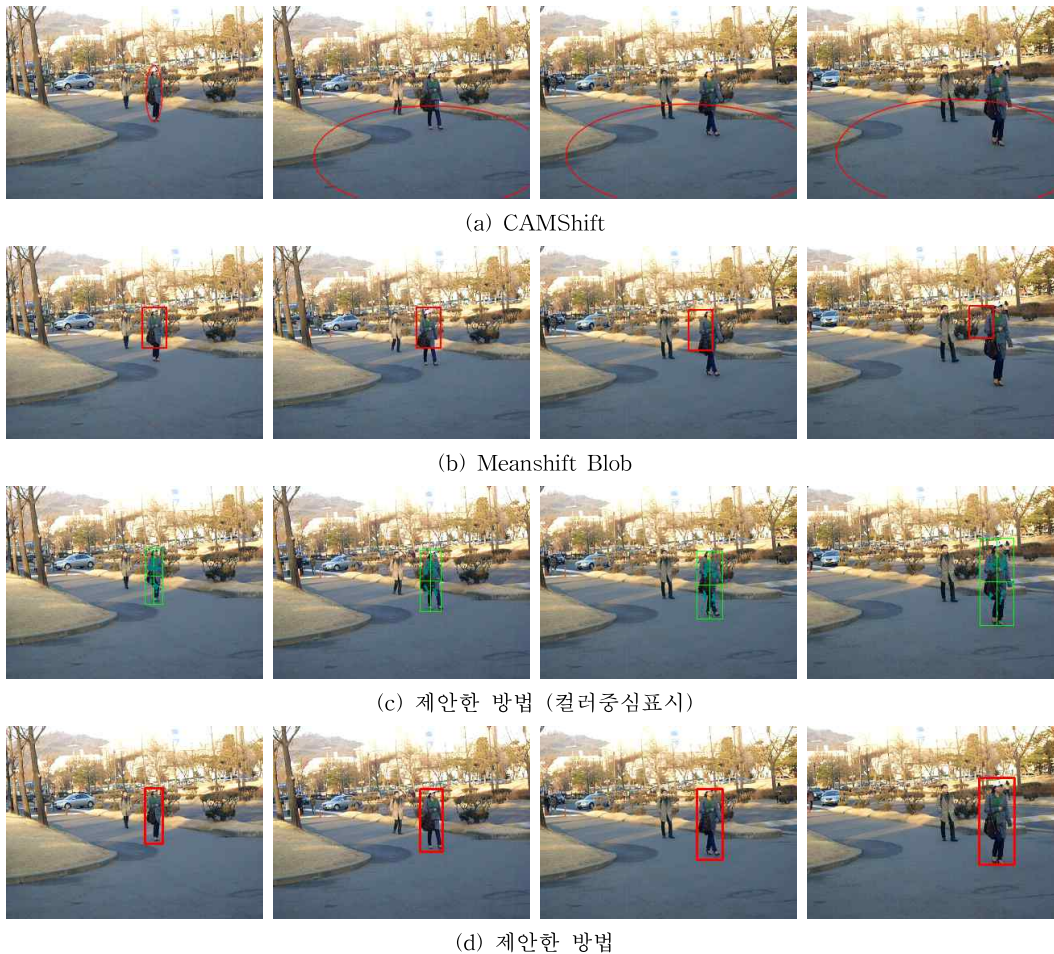


그림 3. 유사한 컬러의 배경이 존재하는 경우의 실험 결과

설명한 스케일 추정방법을 결합시킨 스케일에 적응적인 추적알고리즘의 전체 과정은 다음과 같다.

- ① 초기 프레임에서 표적의 위치와 크기를 설정한다.
- ② 다음 프레임에서 표적영역과 중심은 같지만 크기는 그것보다 약간 큰 표적검색영역을 설정하고 컬러 중심을 구한다.
- ③ 계산된 컬러 중심들을 이용하여 표적의 위치를 계산한다.
- ④ 표적영역을 추정된 위치로 이동시킨다.
- ⑤ 이동된 표적영역을 네 개의 부분영역으로 나누고 각 영역에서 컬러 중심을 구한다.
- ⑥ 각 영역에서의 컬러 중심을 이용하여 모든 줍팩터를 구한다.
- ⑦ 필터링 과정을 이용하여 신뢰할 수 있는 하나의 줍팩터를 구한다.
- ⑧ 최종적으로 계산된 줍팩터를 이용하여 표적영역을 리사이즈(resize)한다.
- ⑨ 다음 프레임을 입력받고 ②~⑧의 과정을 반복 수행한다.

스케일 추정 알고리즘은 표적의 중심을 기준으로 4등분되기 때문에, 추정된 표적의 위치에 따라 그 성능이 크게 좌우된다. 그러므로 2장에서 설명한 컬러 중심의 가중 평균을 이용한 위치 추정과정과의 결합이 최적의 결합이 된다. 왜냐하면 표적영역의 중심이 컬러 중심들의 가중평균이 아니라면, 평균벡터들의 크기가 왼쪽이나 오른쪽, 어느 한쪽으로 기울어질 수 있기 때문에 리사이징이 한쪽 방향으로 편향될 가능성이 높아지기 때문이다.

### 3. 실험결과

제안한 알고리즘의 성능을 실험적으로 증명하기 위하여 평균이동(MeanShift)기반으로 가장 잘 알려진 방법인 CAMShift[7]과 MeanShift Blob[4], 그리고 제안한 방법을 동일한 영상 시퀀스에 적용하여 보았다. 모든 실험에서는 256×256×256의 컬러를 8×8×8로 균일하게 양자화하여 사용하였다. 그림 3은 비교적 복잡하지 않은 배경에 표적이 존재하는 경우를 실험한 결과이다. 배경이 복잡해 보이지는 않지만 표적과 아래 도로에 유사한 컬러가 존재함을 확인할

수 있다. CAMShift의 실험결과는 그림 3(a)에서 보는 바와 같다. 앞에서 언급한 바와 같이 표적 영역을 표시하는 붉은색 타원이 시퀀스가 진행됨에 따라 도로 쪽으로 확산되어 나가는 것을 볼 수 있다. MeanShift Blob의 경우 추적대상의 색상에 대한 분포와 유사한 색상 분포가 원래 추적대상의 크기보다 작은 영역안에서도 존재하기 때문에 표적 윈도우의 크기가 점차로 작아짐을 볼 수 있다. 이에 반해 제안한 방법은 그림 3(c)와 그림 3(d)에서 확인할 수 있는 바와 같이 안정적으로 추적대상의 크기를 계산하고 있음을 확인할 수 있다. 그림 3(c)에서는 표적 영역 내에 실제로 줍팩터를 연산하기 위하여 계산된 각 부분영역에 속한 컬러 중심들을 초록색 점들로 표시하였다.

그림 4는 표적의 주변에 움직이는 여러 배경이 존재하는 경우를 실험한 결과이다. 그림 4(a)에서 보듯 CAMShift는 그림 3의 결과와 마찬가지로 불안정하게 동작하여 추적이 실패함을 알 수 있다. MeanShift Blob 역시 크기를 제대로 추정하지 못하며, 결과적으로 추적이 실패하는 원인이 된다. 이에 반해 제안한 알고리즘은 그림 4(c)와 그림 4(d)에서 보는 바와 같이 추적대상의 크기를 안정적으로 추정함을 확인할 수 있다. 특히 표적이 다른 객체의 의해 부분적으로 폐색(occlusion)되는 경우에도 동작함을 볼 수 있다. 이는 폐색되지 않은 남은 컬러중심들이 크기 추정과 추적을 수행하기 때문이다. 그림 4(c)에서 추적대상이 비록 부분적으로 폐색되더라도 초록색으로 표시되는 컬러중심들이 남아 있음을 확인할 수 있다. 이 컬러중심들에 의해 추적이 가능하게 된다. 그림 5는 추적대상이 정면에서 접근하여 점차 그 크기가 커질 때의 크기 추정에 대한 실험결과이다. 이 경우 크기 변화가 심하고 주위의 컬러들이 있어서 크기 추정이 어려운 경우로 CAMShift나 MeanShift Blob이 모두 실패하며, 그림 5에는 제안한 알고리즘의 결과만을 나타내었다. 제안한 알고리즘의 프레임당 연산시간은 추적대상의 크기와 영상의 크기에 따라 달라지면 추적대상의 크기가 커질 수록 연산시간이 늘어난다. 그러나 기본적으로 위치 추정에 사용된 빈들의 중심들이 크기 추정에 다시 사용되므로 추가적인 계산이 많이 요구되지 않으나 필터링시에 사용되는 정렬에서 대부분의 연산시간이 사용된다. 그림 3의 시퀀스에 대한 평균 연산시간은 프레임당 3.4ms, 그림 4의





(a) CAMShift



(b) MeanShift Blob



(c) 제안한 방법 (컬러중심표시)



(d) 제안한 방법

그림 4. 주변에 복잡한 배경이 있는 경우의 실험결과

시퀀스에 대해서는 4.5ms, 그림 5에 대해서는 7.3ms 정도 걸린다. 이는 실시간으로 추적이 가능함을 보여 준다.

표 1은 제안한 알고리즘에 대하여 검색영역을 다양한 크기로 주었을 때의 중심위치와 추적대상의 크기에 대한 추정 에러를 픽셀단위로 나타낸 것이다.

표 1의 에러 측정을 위해 그림 3, 그림 4, 그림 5에서 쓰인 각 실험 시퀀스들의 30번째 프레임마다 뽑아낸 프레임에 대하여 눈대중으로 추적대상의 위치와 크기를 측정하여 참값으로 사용하였으며, 이 참값과 알고리즘에 의해 추정된 위치 및 크기간의 차이를 에러로 정의하였다. 사용한 프레임수는 20개정도이다. 위



(a) 제안한 방법 (컬러중심표시)



(b) 제안한 방법

그림 5. 추적대상이 정면에서 접근할 때의 크기추정에 대한 실험결과

표 1. 제안한 알고리즘에 대하여 검색영역의 크기에 따른 위치와 크기의 평균적인 오차. 단위는 픽셀임

영상 시퀀스	검색영역의 표적영역에 대한 크기 비율					
	1.2배		1.3배		1.4배	
	평균 위치오차 (단위: 픽셀)	평균 크기오차 (픽셀 갯수차)	평균 위치오차 (단위: 픽셀)	평균 크기오차 (픽셀 갯수차)	평균 위치오차 (단위:픽셀)	평균 크기오차 (픽셀 갯수차)
그림 3	5.6	53.6	6.5	63.6	7.4	71.4
그림 4	6.4	68.5	7.8	73.5	8.8	85.5
그림 5	8.2	82.1	9.3	90.5	11.6	102.5
Cavier Walk	3.2	32.4	3.4	34.3	3.5	36.7

표 2. 검색영역이 추적대상 영역의 1.2배에 달할 때의 위치와 크기의 평균적인 오차의 비교. CAMShift와 MeanShift Blob이 완전히 추적이 실패하기 직전의 프레임까지에 대한 추정결과임.

영상 시퀀스	알고리즘					
	CAMShift		MeanShift Blob		제안한 방법	
	평균 위치오차 (단위:픽셀)	평균 크기오차 (픽셀 갯수차)	평균 위치오차 (단위:픽셀)	평균 크기오차 (픽셀 갯수차)	평균 위치오차 (단위:픽셀)	평균 크기오차 (픽셀 갯수차)
그림 3	24.5	1034.6	19.6	593.6	5.2	51.2
그림 4	28.8	1338.3	23.2	903.4	6.4	65.3
그림 5	36.5	2129.9	31.7	1694.9	8.1	79.6
Cavier Walk	6.5	105.3	5.6	90.2	3.0	30.7

치의 에러는 Ground truth 위치와 측정된 위치간의 Euclidean 거리를 픽셀의 단위로 측정하였다. 크기의 에러는 단순히 면적의 크기의 차이를 픽셀의 단위로 나타낸 것이다. 표 2는 각 알고리즘들에 대하여 위치와 크기의 추정에 대한 평균 에러를 나타내었다. 이때 사용한 프레임들은 CAMShift나 MeanShift Blob이 추적이 크게 실패하기 이전까지의 프레임들을 사용한 것들이다. 표 1과 표 2에서의 Cavier Walk 시퀀스는 추적알고리즘의 비교를 위해 자주 쓰이는 시퀀스로 인터넷상에서 얻을 수 있으며, 이 시퀀스에 대해서는 위치와 크기의 참값이 주어졌기 때문에 추가로 본 시퀀스를 그림으로 나타내지는 않았으나 실험 비교에 사용하였다. 표 2를 통해 제안한 알고리즘이 위치 및 크기 추정을 안정적으로 수행함을 확인할 수 있다.

#### 4. 결 론

본 논문은 컬러들의 중심을 이용하여 표적의 스케일의 추정하는 크기 적응적 추적 알고리즘을 제안하였다. 제안한 방법은 일반적인 방법과 달리 컬러들의

기하학적 중심을 이용하기 때문에 화소의 개수에 그다지 민감하지 않다. 따라서 배경에 유사한 색상이 존재하더라도 비교적 안정적으로 동작할 수 있으며 이를 실험을 통해 검증하였다. 제안한 방법은 중심이 동기반의 위치추적 알고리즘과 결합된 형태로 제시되었다. 그러나 제안된 스케일 추정 방법은 중심이 동기반의 위치 추적 알고리즘 뿐 아니라 기존의 평균이동 추적 방법, 파티클 필터 기반의 추적 방법 등 다양한 추적 알고리즘과의 결합이 가능하다. 따라서 추후 연구를 통하여 제안한 방법을 다른 추적 알고리즘에 적용시킬 것이다.

#### 참 고 문 헌

[1] S. Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms," IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1998.

[2] P. Fieguth and D. Terzopoulos, "Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates," IEEE Int. Conf. on



Computer Vision and Pattern Recognition, pp. 21 - 27, 1997.

[3] S. Benhimane and E. Malis, "Integration of Euclidean Constraints in Template Based Visual Tracking of Piecewise-Planar Scenes," *Intelligent Robots and Systems*, Vol.9, No.15, pp. 1218-1223, 2006.

[4] 황인택, 최색상, "변화를 갖는 객체추적 알고리즘," 한국멀티미디어학회 논문지, Vol.10, No.7, pp. 827-837, 2007.

[5] R. Collins, "Mean-Shift Blob Tracking Through Scale Space," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2003.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.25, pp. 564-575, 2003.

[7] S. H. Lee, E. Choi, and M. G. Kang, "Object Tracking Based on Area Weighted Centroids Shifting with Spatiality Constraints," *IEEE Int. Conf. Image Processing*, San Diego, USA, 2008.

[8] G. R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," *Intel Technology Journal*, 1998.

[9] A. Yilmaz, "Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation selection," *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2007.



이 석 호

1993년 2월 연세대학교 전자공학과 공학사  
 1998년 2월 연세대학교 전자공학과 공학석사  
 2003년 2월 연세대학교 전기전자공학과 공학박사

2003년~2006년 연세대학교 수학과 박사후과정  
 2006년~2008년 연세대학교 전기전자공학과 연구교수  
 2008년~현재 동서대학교 컴퓨터정보공학부 전임강사  
 관심분야: 영상처리, 컴퓨터비전



강 문 기

1986년 2월 서울대학교 전자공학과 공학사  
 1988년 2월 서울대학교 전자공학과 공학석사  
 1994년 2월 Northwestern University 공학박사

1994년 3월~10월 Research Fellow(Northwestern University)  
 1994년 11월~1997년 2월 Assistant Professor (University of Minnesota, Duluth)  
 1997년 3월~1999년 2월 연세대학교 전자공학과 조교수  
 1999년 3월~2004년 2월 연세대학교 전자공학과 부교수  
 2004년 3월~현재 연세대학교 전자공학과 정교수  
 IEEE와 SPIE의 멤버  
 IEEE signal processing magazine의 editorial board member  
 SPIE Milestone Series Volume(CCD and CMOS imagers)의 Editor  
 IEEE ISPCS Steering Committee  
 EURASIP Journal of ASP(Applied Signal Processing), Editor  
 EURASIP Journal of ASP, Special issue on Superresolution, Guest Editor  
 Journal of Advances in Multimedia, Editor  
 관심분야: 영상복원, 초해상도 영상복원, 비선형 필터링, 비디오 분석 및 처리 부화소단위의 움직임 추정