JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 3 : 636~644 Mar. 2015

CAM shift와 8방향 탐색 윈도우를 이용한 객체 추적

김남곤[1] · 이금분[2] · 조범준[1*]

Object Tracking Using CAM shift with 8-way Search Window

Nam-Gon Kim[1] · Geum-Boon Lee[2] · Beom-joon Cho[1*]

[1*]Department of Computer Engineering, Chosun University, Gwang-ju, Korea
[2]Department of Computer Security, Chosun College of Science & Technology, Gwang-ju, Korea

요 약

이 논문에서는 CAM shift알고리즘과 8방향 탐색 위도우를 결합하여 객체의 추적 성능을 향상하는 방법과 추적에 이용되는 프레임의 수를 줄여 연산을 줄이는 방법을 제안한다. CAM shift는 대표적인 색상을 이용한 추적 방법이나 빠른 속도로 이동하는 물체를 추적하기 어려운 단점이 있다. 이를 해결하기 위해 추적 대상을 놓쳐버린 시점에서 마지막으로 추적에 성공한 시점의 정보를 이용하여 8방향 탐색을 실시하여 객체를 찾아 낸 후 CAM shift의 탐색 윈도우를 이동시켜 기존의 CAM shift로는 추적이 불가능한 고속 이동 물체에 대해서도 보다 정확한 추적이 가능하게 되었다. 또한 하드웨어의 발달로 초당 생산되어지는 프레임의 수가 증가하여 불필요한 연산이 증가하게 되었고, 이를 줄이기 위해 추적에 이용되는 프레임의 수를 줄여 연산을 줄여 이 전보다 효율을 높일 수 있었다.

ABSTRACT

This research aims to suggest methods to improve object tracking performance by combining CAM shift algorithm with 8-way search window, and reduce arithmetic operation by reducing the number of frame used for tracking. CAM shift has its adverse effect in tracking methods using signature color or having difficulty in tracking rapidly moving object. To resolve this, moving search window of CAM shift makes it possible to more accurately track high-speed moving object after finding object by conducting 8-way search by using information at a final successful timing point from a timing point missing tracking object. Moreover, hardware development led to increased unnecessary arithmetic operation by increasing the number of frame produced per second, which indicates efficiency can be enhanced by reducing the number of frame used in tracking to reduce unnecessary arithmetic operation.

키워드 : 추적, 탐색, 캠 쉬프트, 탐색 윈도우

Key word : Tracking, Search, CAM shift, Search Window

접수일자 : 2015. 01. 30 심사완료일자 : 2015. 02. 16 게재확정일자 : 2015. 03. 02

* **Corresponding Author** Beom-joon Cho(E-mail:bjcho@chosun.ac.kr, Tel:+82-62-230-7759)
Department of Computer Engineering, Chosun University, Gwang-ju , Korea

Open Access http://dx.doi.org/10.6109/jkiice.2015.19.3.636 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

영상의 디지털화 이후 동영상 분석에 대한 관심이 증대 되었으며, 특정 객체에 대한 추적 연구의 필요성이 증대되고 있다.

이러한 추적은 다양한 분야와 융합되어 적용되고 있으며 대표적으로, 위성을 이용한 위치 추적, 군용화기의 대상 추적, 도심지 내에서의 차량 추적 등 여러 분야에서 목표물에 대한 추적의 요구량이 증대됨에 따라 추적 기법에 대한 연구 또한 활발하게 진행되고 있다.

추적기법 들은 카메라, 센서 등 영상 기기의 발달로 색상, 모양 , 크기, 온도 등 다양한 정보를 영상에서 획득하여 객체를 검출하고 추적해 나간다.

색상 정보를 이용한 대표적인 방법으로 Mean shift와 CAM shift 알고리즘 등이 있다. 특히 CAM shift는 영상 내 색상 확률 분포를 통해 객체를 탐색하고, 영상 모멘트를 통해 탐색 윈도우를 재설정하여 지속적인 추적을 하는 기법으로 최근까지도 색상 기반 연구에 많이 사용되고 있다. 하지만 느리게 움직이는 객체의 위치를 추적할 때 유용하지만 여러 번의 반복을 통해 유추하는 과정과 탐색 윈도우 내의 모든 요소에 대한 연산으로 연산속도가 객체의 움직임을 따라가지 못하는 문제가 발생한다[1-3].

본 논문에서는 연산량의 감소를 위해 프레임 별 간격을 조절함과 동시에 CAM shift를 사용하면서 고속으로 이동하는 물체를 정확히 추적하기 위해 8방향 탐색 윈도우를 이용 하였다.

2장에서는 Mean shift의 발전형인 CAM shift의 특징과 관련연구, 프레임의 간격을 선택적으로 적용하여 얻는 이익에 대해서 소개한다. 3장에서는 8방향 탐색윈도우에 대해 설명하고, 8방향 탐색 윈도우를 CAM shift에 적용할 방법에 대해 설명 하며, 4장에서는 8방향 탐색 윈도우를 적용한 CAM shift의 추적성능을 테스트 한다. 5장에서는 결론을 맺는다.

Ⅱ. CAM shift

고정된 탐색 윈도우를 갖는 Mean shift는 추적이 지속되는 동안 객체의 크기가 변화하여 초기의 크기보다 작아지게 되면 탐색윈도우 내에서 추적 객체의 색상이 차지하는 비율이 줄어들어 배경색이 탐색 윈도우 내의 주 영역이 되어 추적하는 객체를 추적하지 못하게 되는 문제점을 가지고 있다.

이 문제를 해결한 알고리즘이 CAM shift이다. CAM shift알고리즘은 1998년 GR Gradski에 의해 처음 소개가 되었다[4]. 연속되는 영상에서 추적하는 객체의 크기에 따라 탐색윈도우의 크기를 조절하며 추적해 나아가는 것이 특징으로 탐색윈도우의 크기를 조절하여 객체의 비틀림이나 크기 변화에도 추적을 계속 할 수 있게 되었다.

영상의 프레임에서 탐색 윈도우 내 x, y의 색상값을 $I(x,y)$라 하면 초기 상태의 모멘트(Zeroth moment)를 구할 수가 있고 이는 식 (1)로 얻을 수 있다.

$$M_{00} = \sum_x \sum_y I(x,y) \tag{1}$$

x와 y에 대한 첫 번째 모멘트는 식 (2), (3)로 얻는다.

$$M_{10} = \sum_x \sum_y x * I(x,y) \tag{2}$$

$$M_{01} = \sum_x \sum_y y * I(x,y) \tag{3}$$

x와 y의 첫 번째 모멘트를 이용하여 탐색 윈도우 내의 중심x_c와 y_c를 구할 수 있고, 이는 식 (4), (5)와 같다.

$$x_c = \frac{M_{10}}{M_{00}} \tag{4}$$

$$y_c = \frac{M_{01}}{M_{00}} \tag{5}$$

Meanshift는 고정 분포를 갖도록 설계 되어 있다. 하지만 CAMshift는 동적 분포를 가지고 있어 탐색 윈도우의 크기를 능동적으로 바꾸도록 설계 되어있다.

위의 Meanshift의 출력 결과는 CAMshift의 입력 결과로 사용이 되어 질 수 있다. 다음의 식 (6), (7)을 이용하여 두 번째 모멘트를 구할 수 있다.

$$M_{20} = \sum_x \sum_y x^2 * I(x,y) \tag{6}$$

$$M_{02} = \sum_x \sum_y y^2 * I(x,y) \tag{7}$$

구해진 두 번째 모멘트를 이용하여 다음 식 (8),(9), (10)을 계산 할 수 있다.

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \tag{8}$$

$$b = 2\left(\frac{M_{11}}{M_{00}} - x_c * y_c\right) \tag{9}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \tag{10}$$

위의 식 (8), (9), (10)을 이용하여 탐색 윈도우의 너비(l)와 길이(h)를 구할 수 있고 그 식은 다음 식 (11), (12)와 같다.

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \tag{11}$$

$$h = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \tag{12}$$

탐색 윈도우의 크기를 결정하는 또 다른 방법으로 영역에 대한 2차원 색상 분포에서 픽셀의 최대값은 255이므로, 탐색 윈도우의 사이즈는 다음과 같이 구할 수 있다.

$$s = 2 * \sqrt{\frac{M_{00}}{256}} \tag{13}$$

식 (13)을 이용할 경우 탐색 윈도우의 가로와 세로축의 크기가 같은 크기로 구해지기 때문에 본 논문에서는 식(11), (12)의 공식을 이용하여 크기를 구하였다.

2.1. CAM shift의 문제점

CAM shift는 크게 3가지의 문제점을 가지고 있다. 첫 번째로 자체적으로 대상을 선정 할 수 없어 추적할 객체를 포함한 탐색 윈도우의 크기와 위치를 지정해 주어야 한다. 두 번째로 색상 값을 이용한 추적을 하기 때문에 탐색 영역 내의 배경이 객체의 색상과 비슷할 경우 탐색 윈도우가 대상을 추적 할 수 없게 되는 경우가 발생한다. 세 번째로 빠른 속도로 탐색 윈도우를 벗어나는 대상의 경우 추적을 지속 할 수 가 없다[5].

그림 2는 고속으로 이동하는 물체를 놓쳐 버린 이 후 배경을 추적해야 할 객체로 인식해 버린 탐색 윈도우의 모습이다.

첫 번째 문제의 경우 프로그램의 설계 단계에서 추적할 값을 지정하거나 사용자가 상황에 맞추어 직접 지정해 주는 방법이 존재한다.

두 번째 문제의 경우 색상을 제외한 다른 특징 값을 이용하는 추적 방법과의 연계를 통해 극복이 가능하다.

세 번째 문제의 경우 최종 탐색 성공 위치의 주변을 탐색하여 추적 객체를 다시 찾아내어 CAM shift의 탐색 윈도우를 재설정 해주는 방법을 이용 할 수 있다.

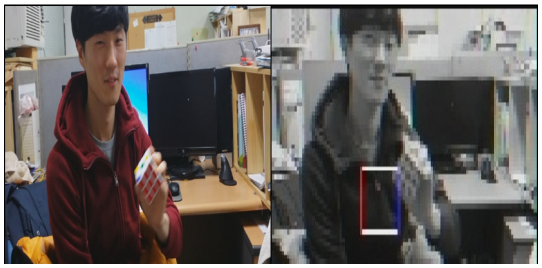


그림 1. CAM shift가 지역 최대점에 빠지는 예
Fig. 1 Example of CAM shift's local maximum

그림 1은 비슷한 배경 색에서 추적해야할 객체와 배경을 구분해 내지 못한 예시 영상 이다.

그림 2. 객체의 급격한 이동 예
Fig. 2 Example of rapid movement

2.2. CAM shift 관련 연구

CAM shift만을 이용해서는 강력한 추적 성능을 구현하기 힘들기 때문에 영상에서 얻을 수 있는 특징 중 CAM shift가 사용하지 않는 특징을 이용한 연구들과 접목되고 있다.

2.2.1. 칼만 필터를 적용한 CAM shift 모델

칼만 필터는 효율적인 무한 임펄스 응답 필터로서, 가우시안 잡음을 가진 선형 동적 시스템에 대한 최적의 예측 방법을 적용하기 때문에 동작 예측 분야에서 가장 널리 알려진 기법 중 하나다.

CAM shift알고리즘과 칼만 필터는 각각 한계점을 가지고 있다. CAM shift알고리즘은 객체를 추적하기 위해 탐색 윈도우를 설정하게 되는데, 이때 설정되는 탐색 윈도우는 이전에 추적하던 객체의 위치를 기반으로 단순이 가로와 세로의 크기를 확대한 것으로 이 탐색 윈도우를 기반으로 다음 객체에 대해 CAM shift를 적용하게 된다. 그래서 이전의 위치에서 급격한 속도로 벗어나는 물체에 대해서는 적용할 수가 없다. 반면 칼만 필터는 객체의 다음 상태를 예측 할 수 있지만, 상태 예측을 위한 값을 칼만 필터가 직접적으로 구하지 않는다. 칼만 필터는 이 측정값을 얻어야만 칼만 필터의 요소들이 갱신되므로 현재 객체에 대한 측정 정보의 계산이 항상 필요하다.

위 두 가지 알고리즘은 서로의 장점으로 서로를 보완할 수 있다. CAM shift의 경우 탐색 윈도우 설정 시 칼만 필터를 이용한 예측 값을 이용하여 창의 위치를 결정 할 수 있고, 칼만 필터의 경우 CAM shift에서 구한 추적 객체의 중심 값을 측정값으로 이용할 수 있다 [6-8].

2.2.2. Depth정보와 SURF알고리즘의 융합 모델

일반적으로 SIFT계열의 알고리즘 방식은 빠른 속도로 움직이는 대상을 놓치기가 쉽다. 그래서 SIFT계열의 속도 부분을 개선시킨 알고리즘이 SURF알고리즘이다. 다만 속도 부분을 개선하여 추출 능력은 SIFT알고리즘 계열 보다 다소 떨어지는 면이 있다.

컬러영상에서 Mean shift와 CAM shift의 사용은 조명과 같은 잡음으로 객체 추적에 실패하는 경우가 있으므로 이를 해결하기 위해 키넥트의 Depth정보를 이용하여 Depth정보가 변화하는 경계선 부분에서 객체를 가우시안 혼합 모델링을 사용하여 분리해내고 분리된 객체의 영역을 관심영역으로 정한 후 SURF알고리즘을 이용하여 특징 점을 추출하고 이런 특징 점을 포함하는 사각형을 만들고 CAM shift를 이용하여 객체를 추적한다[9].

2.2.3. 파티클 필터를 이용한 계층적 모델

파티클 필터에 계층적 모델을 적용한 방법을 제안하였고, 첫 번째 단계에서 컬러의 특징과 HOG(Histogram of Oriented Gradient)특징을 사용하였고 두 번째 단계에서 Haar-like 특징을 사용하고 마지막으로 SIFT방식을 사용하여 목표 객체를 추적 하였다. 비슷한 방법으로 목표의 가려짐이나 추적시간을 줄이기 위해 파티클 필터를 사용하고, Mean shift알고리즘을 사용하여 객체의 위치를 추정하였다[10,11].

2.3. CAM shift의 프레임 간격 조절

최근의 영상 관측 기기 들이 만들어 내는 초당 프레임의 수는 과거의 기기들과는 비교 할 수 없이 증가 하였다. 추적의 실시간성을 보장하기 위해서는 불필요한 연산을 줄일 필요성이 있다. 본 논문에서는 CAM shift의 추적에 사용되는 연산양의 감소를 위해 프레임의 간격을 조절해서 CAM shift를 적용하였다.

표 1. CAM shift의 프레임별 연산양과 추적율
Table. 1 CAM shift's tracking rate and computation rate, per frame

	연산양(단위 : %)	추적율(단위 : %)
1프레임	100	90
2프레임	50	80
3프레임	33	60
4프레임	25	20

표 1은 서로 다른 10개의 영상에서의 프레임 간격별 추적율과 연산양이다.

최초 1프레임간격의 경우 일반적인 CAM shift로 고속으로 이동하는 물체를 추적할 수 없었다. 이후 2프레임에서 4프레임으로 프레임 간격을 늘려가며 실험을 진행하였다. 4프레임 간격의 경우 저속으로 느리게 이동하는 물체는 추적할 수 있었지만, 일반적인 속도로 움직이는 물체는 추적 할 수 없었다.

본 논문의 실험에서는 2프레임 간격으로 프레임을 조절하였다. 표 1의 연산양은 단순 계산으로 다음 식 (14)와 같이 계산한다.

$$\text{연산량} = \frac{\text{계산해야할 프레임의 계산양}}{\text{전체 프레임의 계산양}} \qquad (14)$$

Ⅲ. 8방향 탐색 윈도우

본 논문에서는 2장에서 언급한 고속으로 이동하는 물체를 추적하기 위한 방법으로 8방향 탐색 윈도우를 제안한다. 기존의 방향 탐색 윈도우 들은 지정된 위치에서 설정된 방향으로 객체를 탐색해 간다. 하지만 본 논문에서는 고속으로 이동하여 CAM shift가 놓쳐버린 객체에 대하여 8방향 탐색을 통해 CAM shift가 지속적으로 대상을 추적해 갈 수 있도록 한다.

기존까지의 CAM shift를 이용한 다양한 방법들은 추적 성공률을 올리기 위해 영상의 다음 프레임에서 객체의 위치를 예측하는 방법이 대부분이다.

하지만 위의 방법으로는 갑작스런 진행방향의 변화에 대응하지 못한다는 약점을 가지고 있다. 예를 들어서 칼만 필터를 이용해서 다음 대상의 이동 방향을 예상하여 CAM shift를 이동시키는 방법의 경우 한 방향으로 이동하는 추적 객체에 대해 등가속도 운동 혹은 가속도 운동을 하는 경우 예측이 용이 하다. 하지만 추적하는 대상이 동물이나 사람일 경우 갑작스레 방향을 전환하여 이동하는 경우 대상을 놓치는 경우가 발생하게 된다. 8방향 탐색 윈도우는 대상이 갑작스런 이동속도의 변화와 이동 방향의 전환 시에 CAM shift가 대상을 놓치게 되는 경우 CAM shift보다 넓은 영역에서 대상 객체를 탐색하고 CAM shift의 탐색 윈도우를 탐색된 객체의 위치로 이동시켜 객체를 추적한다.

8방향 탐색 윈도우는 그림 3과 같으며, 8방향 탐색 윈도우의 중심을 추적에 성공한 마지막 중심의 위치로 이동시킨 후 마지막 성공 위치의 평균을 구해 마지막 성공 위치에서 8방향으로 8방향 탐색 윈도우를 이동시켜 각각의 평균값을 구한다. 8방향에서 구한 각각의 평균값과 마지막 성공 위치의 평균값의 차에 절대 값을 취한 값 중 제일 작은 값이 객체의 위치이다.

1	0	0	0	1	0	0	0	1
0	1	0	0	1	0	0	1	0
0	0	1	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	0	0	0
0	0	1	0	1	0	1	0	0
0	1	0	0	1	0	0	1	0
1	0	0	0	1	0	0	0	1

그림 3. 8방향 탐색 윈도우
Fig. 3 8-way search window

객체의 위치를 확인한 후 CAM shift의 탐색 윈도우를 마지막 성공했을 때의 윈도우 사이즈로 조절한 후 8방향 탐색 윈도우를 이용해서 구해낸 위치로 이동 후 추적을 재개 한다. 본 논문에서는 임계값으로 이전의 윈도우 사이즈보다 60%미만으로 줄어들게 되었을 경우 추적을 실패 했다고 상정 하였다. 다음 그림 4는 8방향 탐색 윈도우의 객체 재 추적 의사 코드이다.

Input	$Centers_{n-1}$
Process	`$Calculate\ v_0 \sim v_8$ $v_0 = 8-waywindow(Centers_{n-1})$ $v_{1\sim 8}=8-waywindow$ $(Centers_{n-1} \pm 8wayshift)$ $Centers_n = find$ $(abs(\min(v_0 - v_{1\sim 8})))$
Output	$MoveCAMwindow(Centers_n)$ $ResizeCAMwindow()$

그림 4. 8방향 탐색 윈도우의 객체 재 추적 의사코드
Fig. 4 Pseudocode of 8-way search

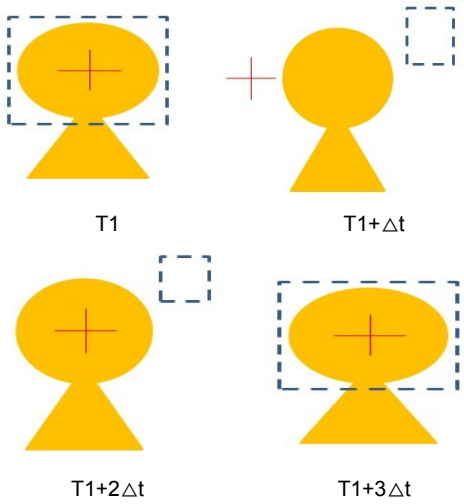


그림 5. 8방향 탐색 윈도우를 이용한 객체 재 추적
Fig. 5 Object research using 8-way search

알고리즘의 진행 순서는 그림 5의 T1과 같이 초기에는 CAM shift만을 반복하며 객체를 추적한다. 추적 중 T1+△t 상태처럼 급격히 객체가 이동한 경우 CAM shift의 탐색 윈도우는 대상을 찾지 못하게 되고 지역 최

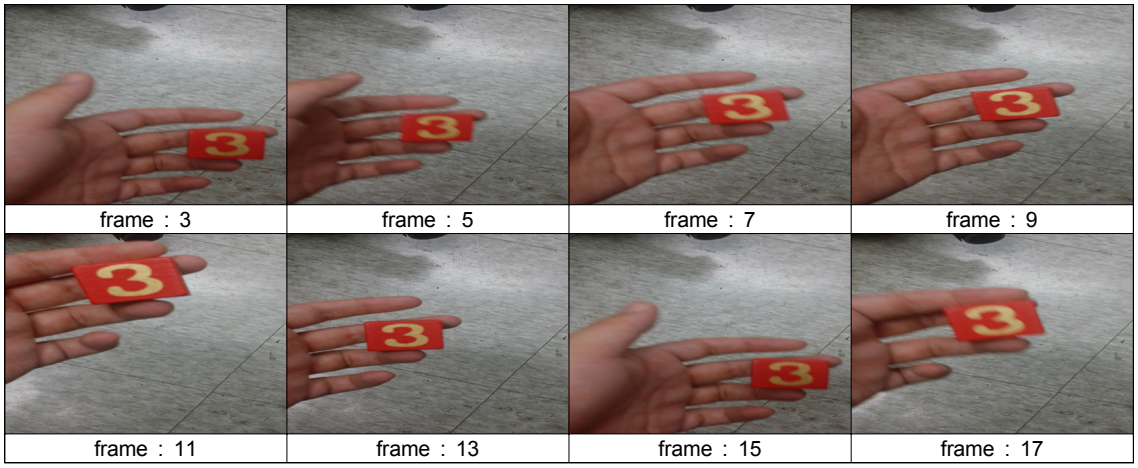


그림 6. 이동 객체 추적 실험 영상
Fig. 6 movement object tracking experimental image

대 점에 자리를 잡게 된다.

T1+△t의 문제를 해결하기 위해서는 이전의 중심 값을 이용하여 다음 중심의 위치를 예측하거나 해당하는 객체를 찾아내야 한다. 하지만 정지 상태에서 급격한 이동을 통해 탐색 윈도우 밖으로 이동하여 버린 객체를 추적하기에는 무리가 따른다. 오랜 시간 정지 상태에 있는 객체의 경우 정지 상태가 다음 객체의 이동 상태로 예측되기 때문이다.

놓쳐버린 객체를 찾아내기 위해서 T1+2△t처럼 마지막에 성공한 추적의 중심에서 8방향 탐색 윈도우를 이용하여 평균값을 계산 한 후 8방향으로 객체를 탐색한다. T1+3△처럼 탐색된 객체의 위치로 CAM shift의 중심점을 옮기고, CAM shift의 탐색 윈도우를 마지막 추적시의 탐색 윈도우의 크기로 설정 한 후 다시 CAM shift를 이용하여 추적을 재개한다.

Ⅳ. 실험 결과

동일한 영상에 대한 CAM shift와 프레임의 간격을 조절한 CAM shift의 객체 추적에 대한 실험을 실시하였다. 그림 13는 추적에 사용한 영상으로 손바닥 위에 붉은 색의 큐브를 올려놓고 큐브에 대한 추적을 실시하였다. 실험에 사용된 모든 영상은 1920 * 1080이며 MATLAB 2010A를 이용하여 실험을 실시했다.

그림 6은 10개의 실험영상중 하나이고, 그림 6은 그림 6의 영상을 이용한 일반적인 CAM shift의 객체 추적이다. 매 프레임 마다 추적을 시행하는 동안 찍힌 중심점을 그래프로 나타낸 것이다. 그림 7는 제안하는 방법대로 그림 5의 영상을 2프레임 간격으로 CAM shift의 추적 간격을 늘려 객체를 추적하였다. 그림 6의 영상을 이용하여 그림 7, 8과 같이 CAM shift의 추적 범위를 멀리 벗어나지 않는 객체에 대한 추적은 문제가 없음을 확인 했다. 대부분의 영상에서도 2프레임 간격의 CAM shift를 이용해서 추적이 가능한지 테스트를 시행하여 보았고, 이것의 결과는 표 1에서 확인 할 수 있다. 2프레임 간격에서 객체의 이동 거리가 멀어 추적 객체가 탐색 윈도우 영역 밖으로 이동하게 되는 경우 고속이동물체를 추적하는 경우처럼 객체를 놓치게 되었다.

다음 그림 10은 고속으로 이동하는 객체에 대한 8방향 탐색 윈도우의 적용 결과 이고, 그림 9는 해당 영상의 프레임별 객체의 위치이다. 실험 영상은 손 위에 추적대상인 큐브를 올려놓고 빠른 속도로 손을 앞으로 이동시켜 일반적인 CAM shift가 추적을 지속하지 못하는 영상이다. 추적 시작 이후 8프레임까지는 문제없이 대상을 추적하는 모습을 확인 할 수 있었다. 하지만 객체의 이동속도가 빨라진 10프레임부터객체를 놓치게 되었다. 하지만 8방향 탐색 윈도우를 적용할 경우 탐색 영역 밖에 있는 객체를 찾아 12프레임에서 추적을 재개 할 수 있었다.

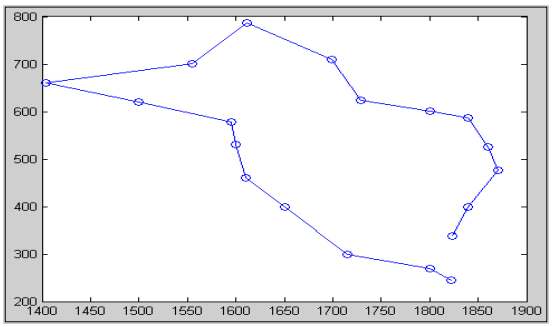


그림 7. 일반적인 CAM shift의 중심점
Fig. 7 General CAM shift's center

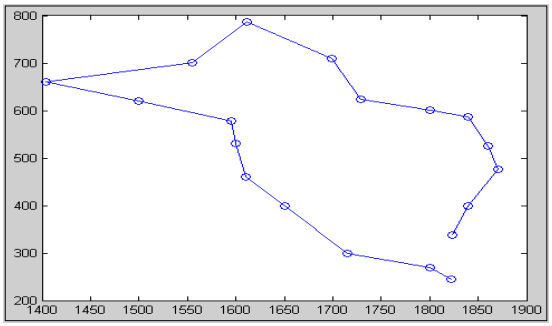


그림 8. 2프레임 간격의 CAM shfit의 중심점의 좌표
Fig. 8 Center of the two-frame interval CAM shift

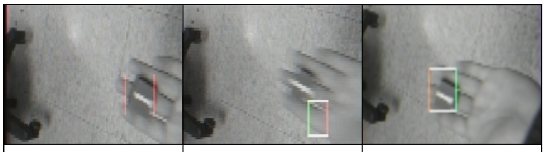


그림 10. 고속 이동 객체 추적
Fig. 10 High speed object tracking‘

실험 중 8방향 탐색 윈도우를 적용하여도 찾을 수 없는 경우가 있었다. 이 경우는 8방향 탐색 윈도우의 탐색 영역 이상으로 객체가 이동해 버린 경우 객체를 추적할 수 없었다. 표 2는 일정속도 이상으로 움직이는 영상에 대한 CAM shift와 8방향 탐색 윈도우를 적용시킨 CAM shift의 추적 성공률의 비교이다.

표 2. CAM shift와 8방향 탐색 윈도우가 적용된 CAM shift의 고속 이동 객체 추적 성공률
Table. 2 The success rate of CAM shift and CAM shift applied 8-way search

	CAM shift	8방향 탐색 윈도우 CAM shift
성공률 (단위 : %)	50	80

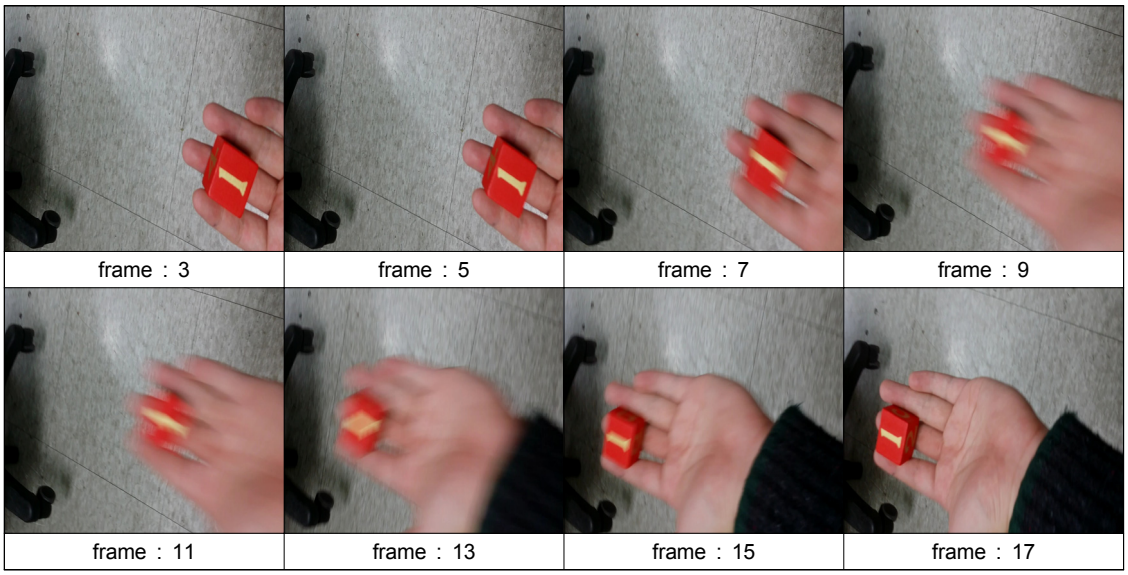


그림 9. 고속 이동 객체 추적 실험 영상
Fig. 9 High speed movement object tracking experimental video image

CAM shift가 추적에 실패하여 객체를 놓치게 되었을 때 8방향 탐색 윈도우의 탐색 영역 내에 추적 객체가 존재할 경우 추적을 계속해 나아갈 수 있었다. 그림 11은 고속으로 이동하는 객체에 대해 제시하는 방법들을 순차적으로 병합한 것으로 1프레임 간격의 CAM shift, 2프레임 간격의 CAM shift, 1프레임 간격의 8방향 탐색 윈도우가 병합된 CAM shift 그리고 2프레임 간격의 8방향 탐색 윈도우가 병합된 CAM shift이다.

첫 번째로 CAM shift가 가지고 있는 문제 점 대로, 1프레임 간격 CAM shift는 고속으로 이동하는 객체에 대해 점차 가속이 붙어 빠르게 움직이는 객체를 놓치게 되었다. 두 번째로 CAM shift가 추적하지 못한 대상에 대해서는 2프레임 간격으로 추적을 시행해도 추적을 실시 할 수 없었다. 세 번째로 1프레임 간격의 CAM shift에 8방향 탐색 윈도우를 병합한 경우 추적 중간 객체를 놓치게 되었지만 객체를 다시 찾아 추적하는 것을 확인 할 수 있었다. 마지막으로 2프레임 간격으로 객체의 추적 간격을 늘려 준 경우 추적 객체를 놓쳤지만 8방향 탐색 윈도우를 이용하여 객체를 재 추적 할 수 있었다.

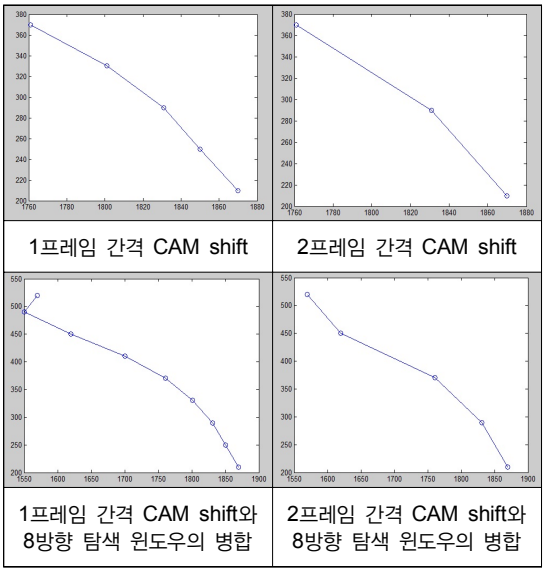


그림 11. 일반적인 CAM shift와 8방향 탐색 윈도우의 병합
Fig. 11 Merge of general CAM shift and 8-way search

Ⅴ. 결 론

CAM shift 알고리즘은 객체의 위치를 추적할 때 탐색 윈도우 내의 모든 요소들을 연산함으로써 계산 비용이 증가되고 빠르게 이동하는 객체 추적을 놓치는 문제가 있었다.

이를 해결하는 방안으로 본 논문에서는 CAM shift 알고리즘과 방향 탐색 윈도우의 병합을 제안하였다. CAM shift 연산량 감소를 위해 프레임의 수를 줄이고자 간격을 두어 프레임을 선택하였으며, 적은 프레임 수의 연산으로 객체 추적이 가능하였다.

그러나 느리게 움직이는 객체를 추적할 때는 추적 성능이 보장되고 계산비용이 감소되었으나 고속으로 움직이는 객체의 경우에는 제대로 추적하지 못하고 객체를 놓치는 오류가 발생하였다. 그래서 놓쳐 버린 객체를 탐색하기 위해 CAM shift의 탐색 윈도우로부터 8방향으로 연산하여 객체가 움직인 위치를 찾아내는 방법으로 CAM shift의 단점을 보완하였다.

이동 객체 추적 실험을 통해 본 논문에서 제안한 방법이 효과적으로 동작함을 결과로 보여주었고, 대내외적인 영상 실험으로 실시간 추적 시스템에서도 적용할 수 있음을 확인하였다.

그러나 초고속으로 움직이는 객체나 영상에서 사라졌다 다시 등장하는 객체의 경우와 같이 탐색 영역 밖으로 나가는 객체 추적의 경우 추적 실패의 문제점이 존재하였다. 따라서 향후 연구 방향은 영상 프레임 간격에 따른 CAM shift 방법의 적용에 있어 영상 내의 추적 객체의 특징을 이용한 중요 프레임만을 선택하여 연산하는 방법과 객체의 이동 속도를 고려한 프레임의 선택, 그리고 8방향 탐색 윈도우를 벗어나 고속으로 움직이는 객체에 대한 추적 방법에 대한 추가적인 연구가 필요하다.

감사의 글

이 논문은 2014년도 조선대학교 학술연구비의 지원을 받아 연구되었음

REFERENCES

[1] P. Vadakkepat, P. Lim, L. C. De silva, L. jing, L. L. Ling, "Multimodal Apporach to Human-face Detection and Tracking", *IEEE Trans. on Industrial Electronics,* Vol. 55, No. 3, 2008.

[2] D Comaniciu and P Meer, "Mean shift: A robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and machine Intellegence*, Vol. 24, No. 5, pp. 603～619, 2002.

[3] D. Exner, E. Bruns, D. Kurz, A. Grundhofer, O. Bimber, "Fast and robust CAMshift tracking", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 9-16, 2010.

[4] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, 2nd Quarter, 1998.

[5] Y.G. Kim, "A Study on Genetic Programming based Generation of the Color Model and Moving Object Tracking using Improved CAMSHIFT Algorithm", SEO KYEONG univ, M.S. dissertation, 2011.

[6] Fang Xu, Jun Cheng, and Chao Wang,"Real time face tracking using particle filtering and mean shift," IEEE *International Conference on Automation and Logic*s, pp. 2252-2255, 2008.

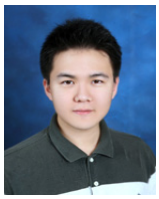
[7] Frat R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface," *Intel Technologt journal*, Vol. 2, No. 2, pp. 12-21 1998.

[8] D. Y. Kim, J. W. Park, C. W. Lee "Object-Tracking System Using Combination of CAMshift and Kalman filter Algorithm," *Journal of Korea Multimedia Society* Vol. 16, No. 5, 2013. 5, pp. 619-628.

[9] J. S. Shin, D. S. Kang "A Study on Moving Object Tracking Algorithm Using Depth Information and SURF Algorithm," *2012 Korean Institute Of Information Technology* pp. 144-148 2012. 5.

[10] C. Yang, R Duraiswami, and L. Davis "Fast Multiple Object Tracking via a Hierarchical Particle Filter," *International Conference on Computer Vision*, Vol. 1, pp. 212～219, 2005. 11.

[11] K. Deguchi, O. Kawanaka, and T. Okatani,"Object tracking by the mean-shift of regional color distribution combined with the particle-filter algorithm," 17th *International Conference on Parttern Recognition*, Vol. 3, pp. 506～509, 2004.



김남곤(Nam-Gon Kim)

2012년 조선대학교 컴퓨터공학과 공학학사
2015년 조선대학교 컴퓨터공학과 공학석사
※관심분야 : 인공지능 패턴인식 패턴인식, 영상처리

이금분(Geum-Boon Lee)

2002년 대전대학교 컴퓨터공학과 공학석사
2010년 조선대학교 컴퓨터공학과 공학박사
2013년~현재 조선이공대학교 컴퓨터보안과 조교수
※관심분야 : 영상보안, 신호처리, 패턴인식

조범준(Beom-Joon Cho)

1980년 조선대학교(B.S., M.S.(82)
1988년 한양대학교 전기공학과 공학박사
2004년 KAIST 전자전산학과 공학박사
1980년~현재 조선대학교 컴퓨터공학부 교수
※관심분야 : 인공지능 패턴인식, 뉴로컴퓨터