creative commons

COMMONS DEED

저작자표시-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

[UCI]I804:24011-200000264551
조선대학교
CHOSUN UNIVERSITY

2015년 2월
석사학위 논문

CAM shift와 8방향 탐색 윈도우를 이용한 객체 추적

조선대학교 대학원

컴퓨터공학과

김 남 곤

조선대학교
CHOSUN UNIVERSITY

2015년 2월 석사학위논문

CAM shift와 8방향 탐색 윈도우를 이용한 객체 추적

김 남 곤

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

CAM shift와 8방향 탐색 윈도우를 이용한 객체 추적

Object Tracking Using CAM shift
with 8-way Search Window

2015년 2월

조선대학교 대학원

컴퓨터공학과

김 남 곤

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

CAM shift와 8방향 탐색 윈도우를 이용한 객체 추적

지도교수 조 범 준

이 논문을 공학석사학위신청 논문으로 제출함.

2014년 11월

조선대학교 대학원

컴퓨터공학과

김 남 곤

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

김남곤의 석사학위논문을 인준함

위원장 조선대학교 교수 김 충 원 (인)

위 원 조선대학교 교수 이 정 아 (인)

위 원 조선대학교 교수 조 범 준 (인)

2014년 11월

조선대학교 대학원

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

목 차

ABSTRACT 7

제1장. 서 론 1

제1절. 연구 배경 및 목적 1

제2절. 연구 내용 및 범위 2

제2장. 관련연구 5

제1절. 색상을 이용한 객체 추적 방법 5

제2절. CAM shift 응용 7

1. 칼만 필터를 적용한 CAM shift 모델 10

2. Depth 정보와 SURF알고리즘의 융합 모델 10

3. 파티클 필터를 이용한 계층적 모델 11

제3장. 제안한 객체 추적 알고리즘 12

제1절. 도입배경 12

제2절. CAM shift알고리즘 12

제3절. CAM shift알고리즘과 8방향 탐색 윈도우의 병합 19

제4장. 실험 결과 24

제1절. 실험 및 결과 분석 24

제5장. 결론 31

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

표 목 차

[표 1] CAM shift의 프레임별 연산량과 추적율 ····· 24
[표 2] 8방향 탐색 윈도우가 적용된 CAM shift의 고속 이동 객체 추적 성공율 ·· 28

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그 림 목 차

[그림 1] CAM shift가 지역 최대점에 빠지는 예 ····· 2
[그림 2] 8방향 탐색 윈도우 ····· 3
[그림 3] HSV 색공간 ····· 6
[그림 4] 객체 추적 시스템 ····· 6
[그림 5] Mean shift 흐름도 ····· 8
[그림 6] CAM shift 흐름도 ····· 9
[그림 7] CAM shift의 객체 추적 ····· 16
[그림 8] 동일 프레임에서 반복적인 중심 이동을 통한 CAM shift의 윈도우 크기 조절 · 17
[그림 9] 추적 객체의 급격한 이동 ····· 18
[그림 10] CAM shift알고리즘과 방향 탐색 윈도우의 병합 ····· 19
[그림 11] 방향 탐색 윈도우를 이용한 객체 재 추적 ····· 21
[그림 12] 방향 탐색 윈도우의 객체 재추적 의사코드 ····· 22
[그림 13] 실험 영상(1) ····· 25
[그림 14] 일반적인 CAM shift의 중심점 ····· 26
[그림 15] 2프레임 간격의 CAM shift의 중심점의 좌표 ····· 26
[그림 16] 고속 이동 객체 추적 ····· 27
[그림 17] 실험 영상(2) ····· 28
[그림 18] 일반적인 CAM shift와 8방향 탐색 윈도우의 병합 ····· 29

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

ABSTRACT

Object Tracking Using CAM shift With 8-way Search Window

Nam-Gon Kim
Advisor : Prof. Beomjoon Cho, Ph.D
Department of Computer Engineering
Graduate School of Chosun University

Accurate tracking for moving objects with CAM shift algorithm requires color information and HUE color values. It also requires an initial search window which is set by the user and it is sensitive to brightness or illumination of the image. There might be a failure in tracking if either a moving object has a high-speed or if it drifts away from the search area for a period of time. Accordingly, it is difficult to keep track of a quick-moving object. The computation cost also increases when all the elements in the search window are computed.

In this paper, we propose an efficient method which uses CAM shift algorithm with 8-way search window. In order to reduce the computation costs, alternate frames were selected in the proposed method. We describe a way to select the frame holding the most important feature of the tracked object in the image and how to select the frame considering the moving speed of the object, and the tracking method of fast-moving objects navigating in 8 directions from the searching window.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제 1장 서 론

제1절 연구 배경 및 목적

영상의 디지털화 이후 동영상 분석에 대한 관심이 증대되었으며 특정 객체에 대한 추적 연구의 필요성이 증대되고 있다. 객체의 추적 분야는 다양한 응용 분야와 융합되어 적용되고 있으며, 위성을 이용한 위치 추적, 군용화기의 대상추적, 도심지 내에서의 차량 추적, 휴먼 추적 등의 여러 분야에서 목표물에 대한 추적의 요구 증대에 따라 추적 기법에 대한 연구 또한 활발하게 진행되고 있다.

최근의 추적 연구는 카메라, 센서 등 영상기기의 발달로 색상, 모양, 크기, 온도 등 다양한 종류의 정보를 획득하여 객체를 검출하고 인식하기 위한 특징으로서 단일 정보가 아닌 정보 간 융합으로 정확한 추적을 지향하고 있다. 그럼에도 동영상 내의 추적 대상은 무작위적이고 빠른 움직임을 가지므로 이를 해결하기 위한 강인한 추적 알고리즘이 요구된다.

색상 정보를 이용한 대표적인 방법으로 Mean shift와 CAM shift 알고리즘 등이 있다. 특히 CAM shift는 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. 영상 내 색상 확률 분포를 통해 객체를 탐색하고, 영상 모멘트를 통해 탐색 윈도우를 재설정하여 지속적인 객체 추적을 하는 기법으로 최근까지도 색상 기반 객체추적 연구에 많이 사용되고 있다[1, 2]. 그러나 CAM shift 알고리즘의 경우 i) 사용자에 의한 초기 탐색 윈도우 설정이 필요하며, ii) 색상 정보를 기반으로 하기 때문에 영상의 조명이나 밝기에 민감하고, iii) 빠르게 이동하는 객체의 경우 탐색 윈도우를 벗어나 지속적인 추적의 어려움을 갖게 된다[3].

본 논문에서는 이러한 CAM shift을 사용하면서 정확한 객체 추적을 하고자 밝기 변화와 색상 차이에 덜 민감한 HUE 색상 값을 사용하며, 고속으로 이동하는 객체의 지속적인 추적을 위해 8방향 탐색 윈도우를 제안한다. 먼저 객체의 움직임과 가변적인 크기에 따른 추적은 CAM shift를 사용하고, 빠르게 움직이는 객체를 추적하는 경우 8방향의 탐색 윈도우 연산으로 정확한 추적을 가능케 한다. 따라서

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

본 논문은 두 가지 알고리즘을 사용하여 빠르게 움직이는 목표물의 정확한 추적이 가능하도록 하였다.

제 2절 연구 내용 및 범위

본 논문에서는 CAM shift가 가진 장점인 빠른 연산을 극대화시키기 위해 효율적인 프레임 간격을 실험을 통해 산출하고 CAM shift의 문제점인 탐색 영역 밖의 객체를 추적하지 못하는 점을 해결하기 위해 8방향 탐색 윈도우를 이용하였다.

CAM shift는 크게 3가지의 문제점을 가지고 있다. 첫 번째로 자체적으로 대상을 선정 할 수 없어 추적할 객체를 포함한 탐색 윈도우의 크기와 위치를 지정해 주어야한다. 이 문제는 추적해야할 객체가 탐색 윈도우의 배경보다 작아서는 안 되고 탐색 윈도우의 대부분의 성분을 추적할 객체가 차지해야 한다. 즉, 탐색 윈도우 내의 색상 히스토그램을 이용하여 추적을 하는 CAM shift의 경우 탐색 윈도우 내의 대부분을 추적할 객체가 차지하지 못할 경우 추적해야할 객체보다 배경이 주된 성분이 되어 추적을 할 수 없다.

두 번째로 HSV 색상 값을 이용한 CAM shift는 탐색 영역 내의 주 색상을 추적하게 된다. 그렇기 때문에 추적하는 영상에서 탐색 영역의 배경과 비슷한 색상의 값을 가진 배경이나 사물이 나타날 경우 그림 1과 같이 CAM shift의 탐색 윈도우가 추적할 대상을 혼동하게 되는 경우가 생기게 된다.

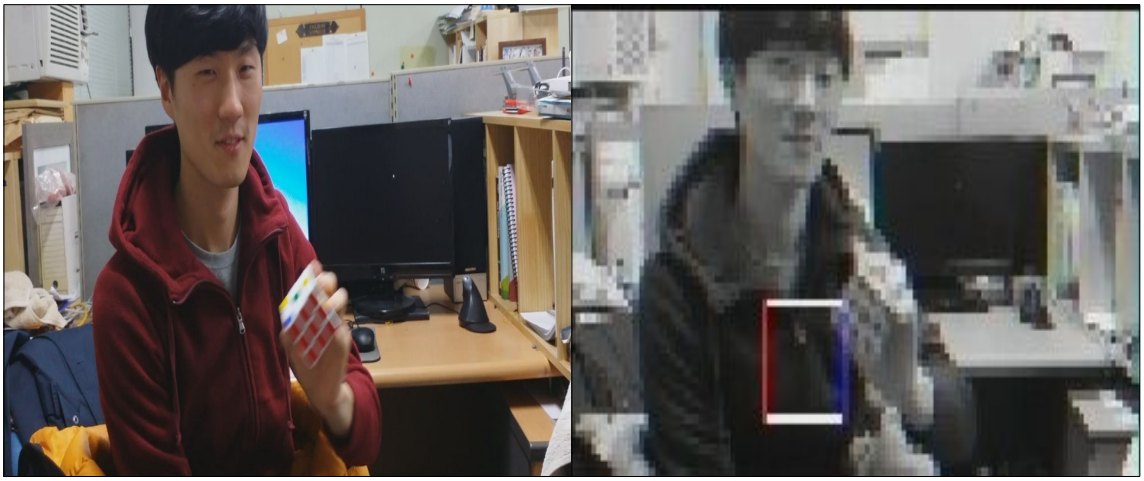


그림 1. CAM shift가 지역 최대점에 빠지는 예

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

세 번째로 빠른 속도로 탐색 윈도우를 벗어나는 대상의 경우 추적을 지속 할 수가 없다. 이전 프레임에서 이후의 프레임으로 넘어가는 단계에서 CAM shift의 탐색 윈도우의 대부분의 영역에 추적할 객체가 포함 되어 있지 않을 경우 탐색 윈도우 내의 색상의 대부분의 값이 배경의 값이 되어버리기 때문이다.

첫 번째 문제의 경우 프로그램의 설계 단계에서 추적할 값을 지정하거나 사용자가 상황에 맞추어 직접 지정해 주는 방법 외에는 해결의 방법이 존재하지 않는다.

두 번째 문제의 경우 색상을 제외한 다른 특징값을 이용하는 추적 방법과의 연계를 통한 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. 혹은 이전 CAM shift의 중심 위치의 값을 알고 있을 경우 다음 위치의 값을 예측해 낼 수 있는 칼만 필터와 같은 알고리즘과의 연계를 통해 극복 할 수 있다. 하지만 이전 중심 값이 존재 하지 않거나 한 위치에 장시간 있을 경우 예측값이 정지값으로 바뀌기 때문에 좋은 방법이라고 할 수가 없다.

본 논문에서는 세 번째 문제의 해결을 위해서 탐색윈도우 밖으로 이동한 객체를 재 추적하기 위해서 8방향 탐색 윈도우를 이용하여 놓쳐버린 객체의 위치를 추적한다.

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 2. 8방향 탐색 윈도우

8방향 탐색 윈도우는 그림 2와 같으며, 8방향 탐색 윈도우의 중심을 추적에 성공한 마지막 중심의 위치로 이동시킨 후 평균을 구한 후 마지막 성공 위치에서 8방향으

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

로 8방향 탐색 윈도우를 이동 시킨 후 각각의 평균값을 구한다. 8방향에서 구한 각각의 평균값과 마지막 성공 위치의 평균값의 절대값을 취한 값 중 최소값이 객체의 위치이다. 이후에는 CAM shift의 탐색 윈도우를 마지막 성공했을 때의 윈도우 사이즈로 조절한 후 8방향 탐색 윈도우를 이용해서 구해낸 위치로 이동 후 추적을 재개 한다.

추가적으로 다양한 촬영 센서의 발달로 단위 시간당 촬영하는 프레임이 증가하게 되므로 모든 프레임을 계산하는 CAM shift의 경우 불필요한 연산량이 생길 수가 있다. 그러므로 연산량 감소를 꾀하면서도 추적의 성능을 유지할 수 있는 간격으로 CAM shift의 연산 간격을 조정 하여 준다.

본 논문의 구성은 다음과 같다. 1장 서론에 이어 2장에서는 기존에 연구되었던 CAM shift에 대해 소개하며, 3장에서는 본 논문에서 제안하는 CAM shift와 8방향 탐색 윈도우를 이용한 추적에 대해 설명하고, 4장에서는 적용한 실험 결과를, 5장에서는 실험 후 남은 향후 과제와 과제에 따른 결론을 이야기 하며 글을 맺는다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

제 2장 관련연구

현재 널리 쓰이고 있는 CAM shift는 Mean shift의 발전형으로 추적 객체의 크기와 뒤틀림의 변화에 대응 할 수 없었던 Mean shift의 약점을 보완하였으나, 고속 이동 객체와 급격한 크기 변화에 완벽하게 대응 할 수 없었기 때문에 다른 여러 방법들과 접목되어 사용된다. 대표적으로 칼만필터나 파티클필터와 결합된 모델, 향상된 칼만필터를 이용한 모델[15], 키넥트를 이용해 Depth정보를 반영하고 SURF(Speeded Up Robust Feature)알고리즘의 특징을 추출해 접목한 CAM shift[8] 등 다양한 방법론이 적용되어 왔고, 이러한 연구는 색상만을 이용하여 추적하는 Mean shift계열의 추적 방법이 가지는 약점을 보완 할 수 있게 하였다.

제 1절 색상을 이용한 객체 추적 방법

Mean shift 알고리즘은 RGB값을 HSV(색상(Hue), 채도(Saturation), 명도(Value))로 변환하고 이중 색상값을 이용하여 지정된 영역내의 객체를 추적하는 방법으로 본 논문에서 사용한 CAM shift 또한 색상값을 이용해서 객체를 추적한다. 색상은 그림 3처럼 가시광선 스펙트럼을 고리모양으로 배치한 색상환에서 가장 파장이 긴 빨강을 0°로 하였을 때 상대적인 배치 각도를 말하며 값은 0° ~ 360°의 범위를 가지고 있으며 0°와 360°은 같은 빨강색을 가리킨다.

추적 알고리즘은 그림 4와 같은 순서를 갖게 된다. 추적하고자 하는 객체가 촬영되어있는 영상을 프레임 별로 구분하고, 구분된 영상에서 관심 영역(Region Of Interest)을 선정한 후, 해당 영역에서 추적에 사용할 정보에 알맞은 데이터를 추출해 낸다.

관심 영역에는 대상의 형태, 질감, 색상 등의 다양한 정보가 있다. 본 논문에서 사용한 CAM shift는 관심 영역의 색상 값을 이용하여 색상 값의 중심을 구하고 그 중심으로 탐색 윈도우의 중심을 이동 시킨 후 탐색을 계속하는 Mean shift에 윈도우의 크기를 갱신하는 것을 추가한 것이다[5].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

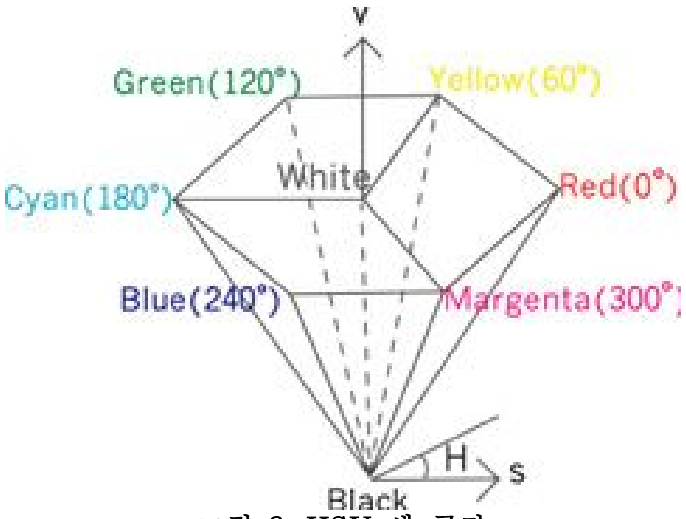


그림 3. HSV 색 공간

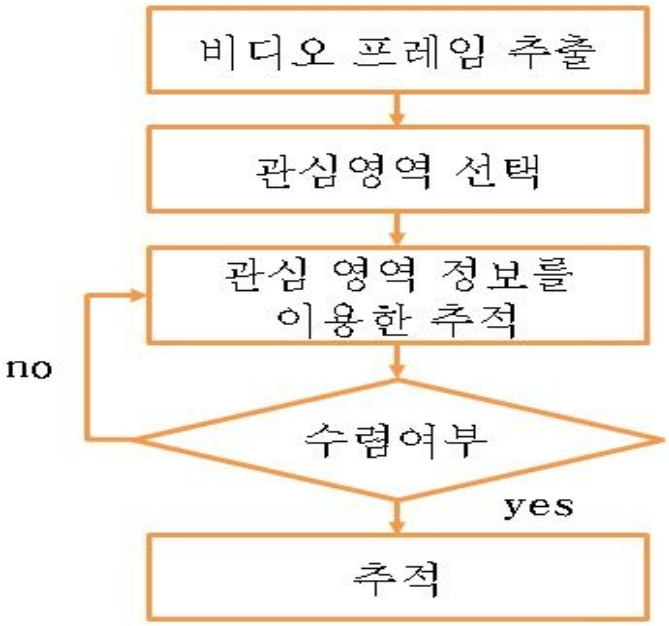


그림 4. 객체 추적 시스템

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

기본적인 Mean shift 알고리즘은 탐색 윈도우의 크기를 갱신하지 못해 지역 최대값에 수렴하여 원하는 객체를 놓치는 경우가 생기게 되어 이를 보완하기 위해 만들어 진 것이 CAM shift알고리즘 이다[5, 6].

여기서 말하는 지역 최대값은 빠르게 움직이는 대상이나 크기의 변화가 있는 대상을 추적 할 경우 탐색 윈도우에서 대상을 놓치게 되어 놓쳐버린 지점이나 이전에 가지고 있는 값과 유사한 특정 지점에서 탐색을 멈추게 되는 경우를 말한다.

위의 지역 최대점에서 추적을 멈추는 현상을 개선한 것이 CAM shift로 Mean shift에 탐색윈도우의 크기를 늘리고 줄이며 대상을 추적해 가는 것을 반복함으로써 지역 최대값에 수렴하는 것을 방지 한다.

그림 3은 본 논문에서 제시하는 알고리즘의 흐름도이다. CAM shift의 탐색 윈도우가 추적 객체를 추적 중 객체를 놓치게 될 경우 8방향 탐색 윈도우가 마지막 탐색 성공 위치에서 평균값을 이용한 추적을 시행 하게 되고 객체를 추적에 성공한 경우 다시 CAM shift의 탐색 윈도우의 크기와 위치를 조절하여 추적을 재개 한다.

제 2절 CAM shift 응용

Mean shift는 데이터 집합의 밀도분포를 이용하여, 객체를 고속으로 추적하는 알고리즘으로 초기 탐색 윈도우의 크기와 위치를 지정하고 색상 값을 색 분할 계산으로 색 클러스터를 발생시켜 초기 지정 영역의 색상에 기반을 두어 탐색 윈도우 내의 객체를 추적하게 된다[6, 14].

하지만 탐색윈도우의 크기를 갱신 하지 못해 추적 대상의 크기가 변하거나 뒤틀릴 경우 지역 최대값에 수렴되기가 쉽고, 이러한 문제를 해결하기 위해서 탐색 윈도우의 크기를 능동적으로 바꿀 수 있는 방법이 필요했고 1998년 GR Gradski에 의해 처음으로 Mean shift의 발전형인 CAM shift가 소개가 되었다[5]. 이 방법을 통해서 능동적으로 변화하는 객체의 크기를 검출하고 추적해 낼 수가 있었고, 다양한 상황에서 카메라의 위치 변화 등에도 적응 할 수 있었다.

Mean shift의 흐름은 그림 3과 같고 CAM shift의 흐름은 그림 4와 같다. 여러

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

연구에도 불구하고 색상값을 이용하는 SIFT(Scale Invariant Feature Transform) 계열의 방법만을 가지고는 비슷한 배경색 위를 지나가는 대상객체를 정확하게 분리하고 추적 할 수 없기 때문에 다른 방법론과 접목되어 사용 되어 진다.

다음 그림 5, 6의 수렴은 추적 시 해당 프레임에서의 반복적인 연산중 이전의 탐색 윈도우의 평균값과 유사도가 높아 추적 객체를 잘 추적하고 있을 때를 말한다.

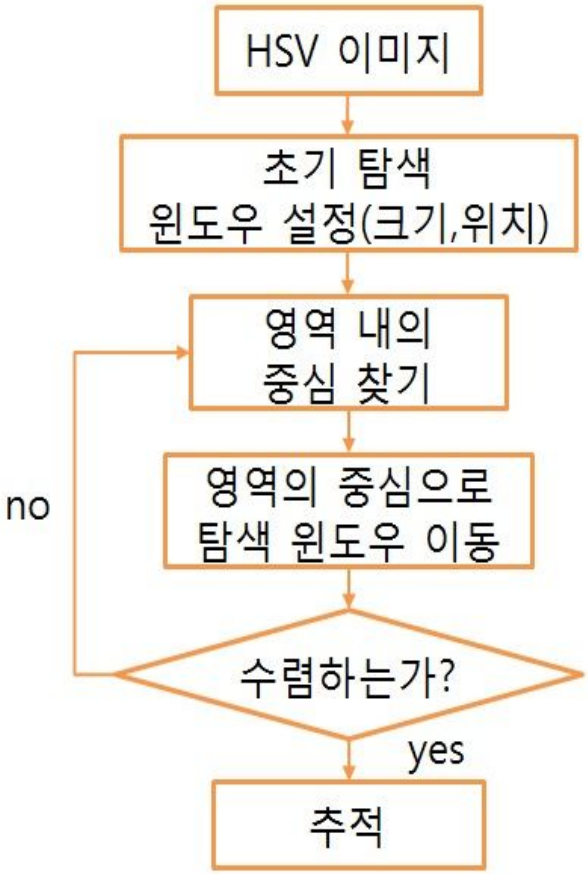


그림 5. Mean shift의 흐름도

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

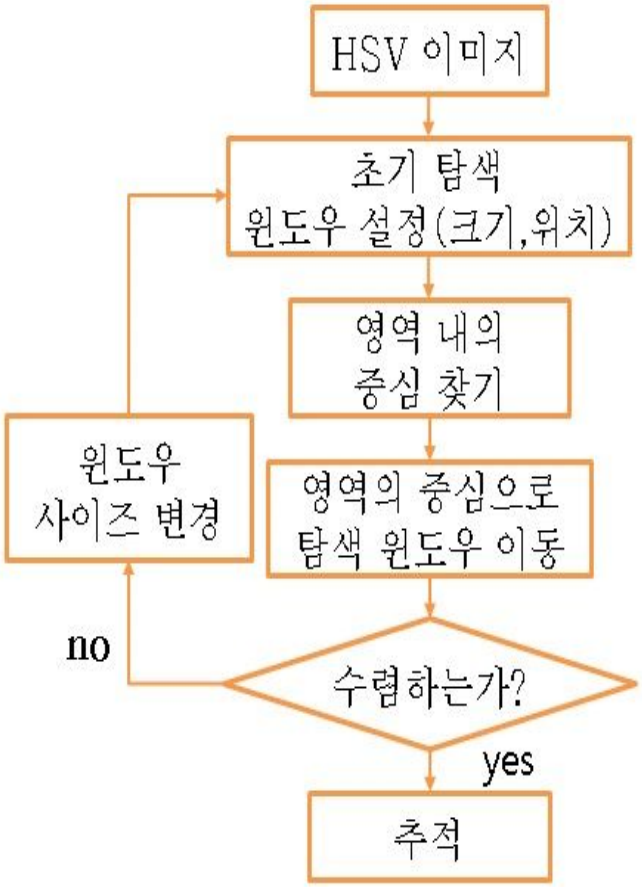


그림 6. CAM shift의 흐름도

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

1. 칼만 필터를 적용한 CAM shift 모델

칼만 필터는 효율적인 무한 임펄스 응답 필터로서, 가우시안 잡음을 가진 선형 동적 시스템에 대한 최적의 예측 방법을 제공하기 때문에 동작 예측 분야에서 가장 널리 알려진 기법중 하나이다[7].

CAM shift알고리즘과 칼만필터는 각각 한계점을 가지고 있다. CAM shift 알고리즘은 객체를 추적하기 위해 탐색 윈도우를 설정하게 되는데, 이때 설정되는 탐색 윈도우는 이전에 추적하던 객체의 위치를 기반으로 단순이 가로와 세로 크기를 확대한 것으로 이 탐색 윈도우를 기반으로 다음 객체에 대해 CAM shift를 적용하게 된다. 그래서 이전의 위치에서 급격한 속도로 벗어나는 고속이동 객체에 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 알고리즘의 장점이 서로를 보완 할 수 있다. CAM shift의 경우 탐색 윈도우 재설정 시 칼만 필터를 이용한 예측 값을 이용하여 창의 위치를 결정할 수 있고, 칼만 필터의 경우 CAM shift에서 구한 추적 객체의 중심 값을 측정값으로 이용할 수 있다[8, 11].

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 특징점을 추출하고 이런 특징점을 포함하는 사각형을 만들고

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

CAMshift를 이용하여 객체를 추적한다[4].

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[10]. 비슷한 방법으로 목표의 가려짐이나 추적시간을 줄이기 위해 파티클 필터를 사용하고, Mean shift알고리즘을 사용하여 객체의 위치를 추정하였다[12].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제 3장 제안한 객체 추적 알고리즘

제 1절 도입배경

인류의 발달에 따라서 선사시대에 식량을 얻기 위해 동물을 추적 하던 단순한 추적의 개념이 디지털로 옮겨와 원하는 객체를 추적하는 단계에 다다랐다. 이제는 위성, CCTV등 다양한 관측 기기로 원하는 객체를 추적하기 위한 방법들이 연구 되고 있으며 객체 추적은 크게 두 가지 8방향으로 연구가 진행 되어 지고 있다. 첫 번째로 결정적 방법(Deterministic method)은 전형적으로 템플릿영상과 현재 영상 사이의 유사 비용 함수의 지역적 극대점을 여러 횟수 반복해 찾는 방법으로 추적을 수행한다. 이 방법을 사용하기 위한 비용 함수로는 최소제곱오차(SSD : Sum of Squard Differences)와 Mean shift알고리즘이 주로 사용 되어 진다.

이와 다른 두 번째 통계적 방법(Stochastic method)은 추적 시스템의 다양성을 모델링하기 위해서 상태공간을 이용한다. 일반적으로 연속적인 확률에서 다음 상태를 구해내는 파티클 필터를 이용한 방법이 있다.

위의 두 가지 방법을 각기 사용하기도 하지만 서로의 장점만을 취해서 적용하는 경우도 많으며, 방법론을 적용한 시도들이 지금도 이루어지고 있다.

본 논문에서는 하드웨어 적인 기기들의 발달에 맞추어 CAM shift의 연산량을 줄이고 추적하는 객체를 놓치게 되었을 때 8방향 탐색 기법을 이용한 객체의 추적을 제안한다.

제 2절 CAM shift알고리즘

CAM shift는 Mean shift의 흐름에 탐색 윈도우 사이즈의 조절 부분이 추가된 것으로 다른 부분의 흐름은 Mean shift의 흐름을 따른다.

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

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

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

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_c와y_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.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

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

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

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.

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

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

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

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.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

앞서 말한 CAM shift 알고리즘은 색상을 이용하여 객체를 추적 하며 대부분의 경우 큰 무리 없이 객체를 추적 하게 된다. 하지만 추적 대상인 객체가 급격하게 가속하여 탐색 윈도우 밖으로 나가게 될 경우 그림 7과 같이 객체를 추적 할 수 없게 된다.

T1 : 초기 상태의 CAM shift

T1+△t : 추적 상태의 CAM shift

T1+2△t : 객체의 급격한 이동으로 추적에 실패한 CAM shift

그림 7. CAM shift의 객체 추적

T1은 초기 상태의 CAM shift의 모습으로 추적할 객체에 알맞은 윈도우와 위치를 설정해 준 상태이다. T1+△t는 추적을 시작한 CAM shift가 대상을 추적중인 상태로 탐색 윈도우 내의 색상의 중심으로 탐색 윈도우의 중심을 옮겨가며 객체를 추적해 간다. T1+△t에서는 지정된 횟수 혹은 지정된 임계값에 다다를 때까지 탐색윈도우의 중심을 탐색윈도우 내의 색상의 중심으로 이동시키게 된다. 즉, 한 프

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

레임에서 반복하는 횟수를 한 회라고 할 때, 여기서 탐색 윈도우 내에서 일정량만큼 추적 객체가 이동하였다면 한 회에서 객체를 추적 하는 방법은 반복 횟수 동안 지속적으로 탐색윈도우를 무게중심의 위치로 이동시키고 탐색 윈도우의 크기를 조절한 것이다. 그림 8은 동일한 프레임에서 윈도우 사이즈를 조절하며 추적을 하는 CAM shift의 모습이다. 좌측의 탐색 윈도우보다 우측의 탐색 윈도우가 더 큰 것을 확인 할 수 있다.

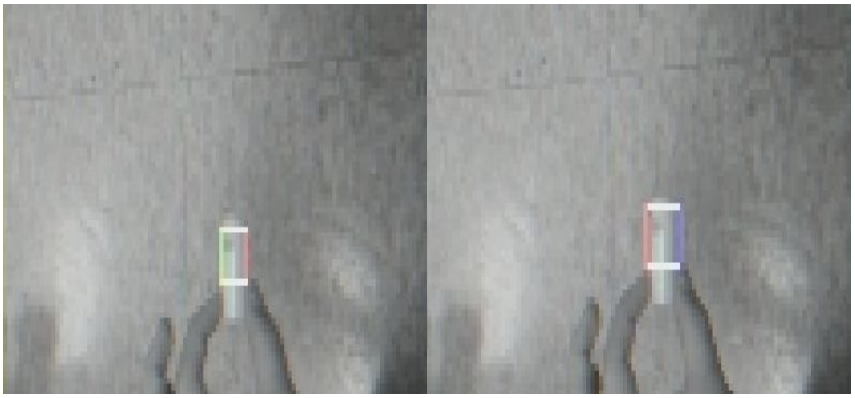


그림 8. 동일 프레임에서 반복적인 중심 이동을 통한 CAM shift의 윈도우 크기 조절

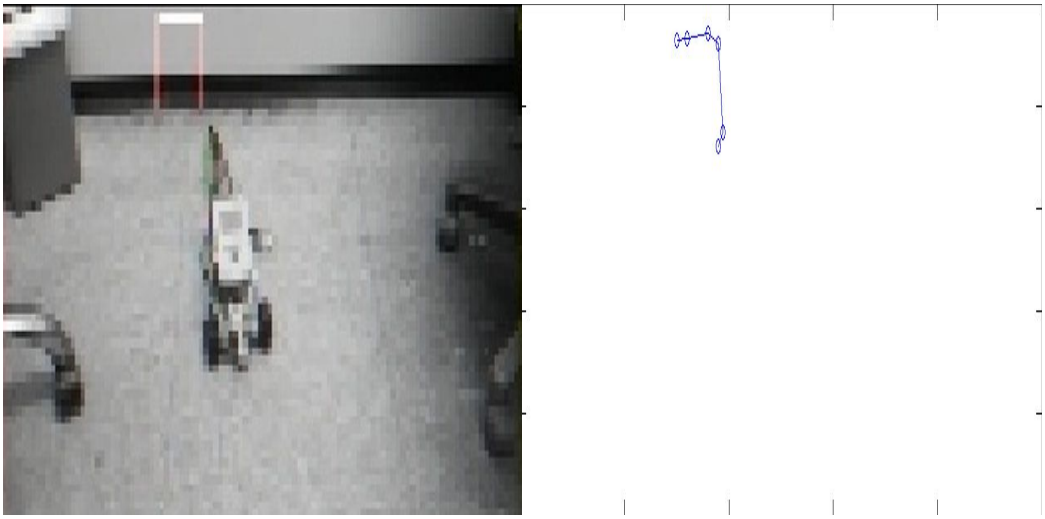
다음으로 T1+2△t처럼 추적 객체가 급격한 속도로 가속하여 탐색 윈도우 밖으로 나가게 된 경우 영역내의 색상값을 이용하여 추적을 하는 CAM shift의 경우 탐색을 지속할 수 없게 된다.

그림 9은 급작스런 추적 객체의 가속으로 추적 대상을 잃어 버렸을 경우의 모습으로 영상의 2프레임과 6프레임의 모습이며, 우측의 그래프는 각각 2프레임과 6프레임의 탐색 윈도우의 중심점을 순서대로 연결한 모습이다. 추적 객체가 급격하게 이동하여 탐색 윈도우 밖으로 나가게 될 경우 CAM shift는 원래 추적하는 대상과 비슷한 색상 값의 위치로 이동해 멈추는 것을 확인 할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

frame 2



frame 6

그림 9. 추적 객체의 급격한 이동

이러한 상황을 해결하기 위해서는 탐색 윈도우 밖으로 나가버린 객체를 다시 찾아낼 방법이 필요하다.

앞서 언급한 관련연구들은 CAM shift의 약점을 보완하기 위해서 객체의 이전 이동 값을 이용하여 앞으로의 이동 위치를 예측해서 CAM shift의 중심점의 위치를 예측 값으로 이동시키는 방법, 다른 객체 혹은 그림자와 겹쳐질 경우 추적하는 객체와 배경을 분리 하여 추적을 하는 방법, 추적의 성능을 높여 추적중인 객체를 놓칠 확률을 낮추는 방법들이기 때문에 급격한 이동을 통해 탐색 윈도우 밖으로

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

나가버린 객체에 대해서는 이전의 정보가 불확실하게 되어 추적 방법으로는 적절하지 않다[7-12].

제 3절 CAM shift알고리즘과 방향 탐색 윈도우의 병합

본 논문에서는 고속으로 이동해서 탐색 윈도우 밖으로 나간 객체를 추적하기 위한 방법으로 8방향 탐색 윈도우의 사용을 제안 한다.

그림 10은 8방향 탐색 윈도우를 적용한 CAM shift의 흐름도이다.

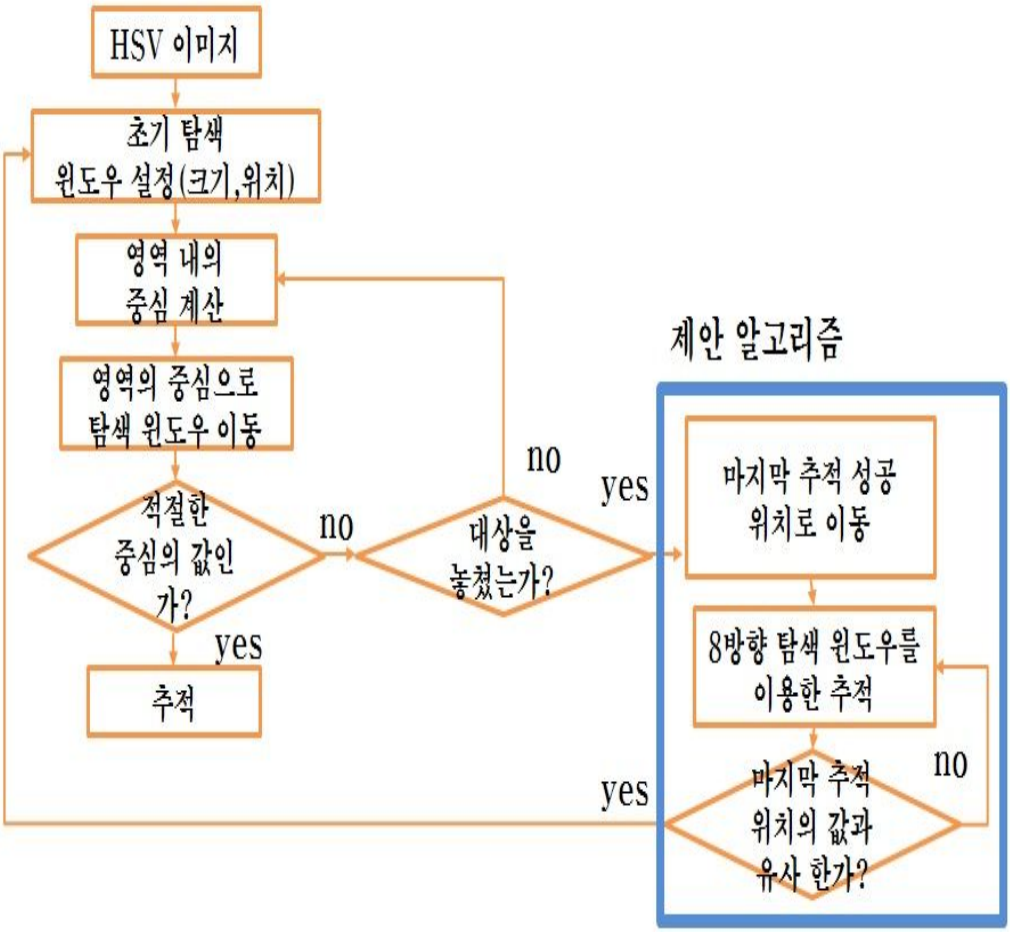


그림 10. CAM shift알고리즘과 방향 탐색 윈도우의 병합

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

객체를 추적 중 객체를 놓치게 될 경우 대부분의 경우 CAM shift의 탐색 윈도우는 수축하게 된다. 본 논문에서는 임계값으로 이전의 윈도우 사이즈보다 60%미만으로 줄어들었을 경우 추적을 실패 했다고 상정 하였다.

객체를 놓치게 되면 8방향 탐색 윈도우는 마지막 추적 성공 위치의 중심점으로 이동하여 탐색 윈도우의 값을 계산하게 되고 이후 8방향의 값을 각각 계산하여 마지막 추적이 성공한 값과 유사도를 측정한다. 계산된 값 중 마지막 추적 성공 위치의 값과 가장 유사한 값으로 CAM shift의 탐색 윈도우를 이동시키고 마지막 성공 시의 CAM shift의 탐색 윈도우의 크기로 재설정 하여 준다.

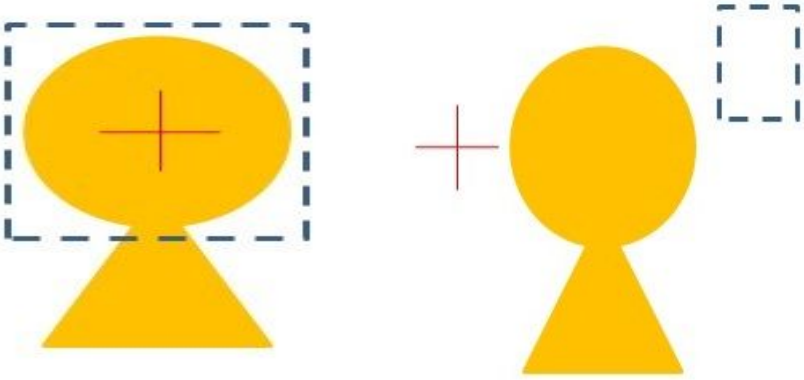
본 논문에서는 하드웨어 센서의 발달에 따라 생성되는 영상의 초당 프레임이 증가하였으며, CAM shift를 이용한 추적에 필요한 프레임 수보다 많은 프레임이 생산되고 있어 연산량 줄이기 위해 CAM shift가 추적에 사용하는 프레임의 간격을 조절하여 보았다.

연산량을 구하기 위한 방법은 1 프레임 단위로 추적을 하였을 경우를 연산량 100%로 놓고 프레임 간격을 늘려서 추적을 할 경우 연산에 할애해야 하는 연산량은 식 (1)과 같다.

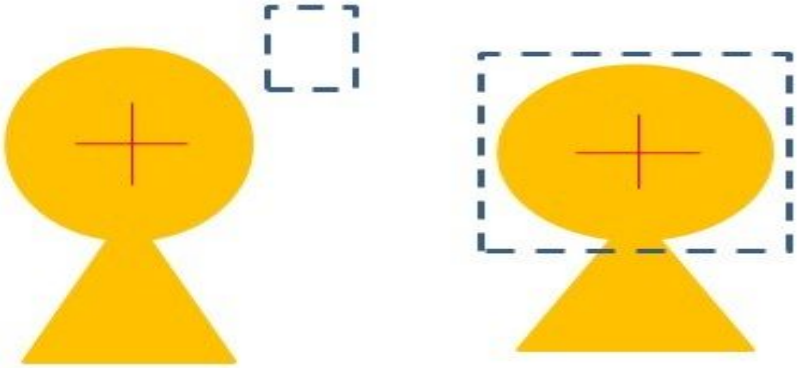
$$\text{연산량} = \frac{\text{계산해야할 프레임의 계산량}}{\text{전체 프레임의 계산량}} \tag{1}$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY



T1 : CAM shift　　T1+△t : 추적 대상을 놓친 상태



T1+2△t : 8방향 탐색 윈도우를 이용한 객체 추적

T1+3△t : CAM shift의 중심을 8방향 탐색 윈도우의 중심으로 이동

CAM shift 탐색 윈도우　8방향 탐색 윈도우

그림 11. 방향 탐색 윈도우를 이용한 객체 재 추적

알고리즘의 진행 순서는 그림 11의 T1과 같이 초기에는 CAM shift만 탐색을 반복하며 객체를 추적한다.

추적 중 T1+△t 상태처럼 급격히 객체가 이동한 경우 CAM shift의 탐색 윈도우는 대상을 찾지 못하게 되고 지역 최대 점에 자리를 잡게 된다. T1+△t의 문제를 해결하기 위해서는 이전의 중심 값을 이용하여 다음 중심의 위치를 예측하거나 해

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

당하는 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 기법의 윈도우를 이용하여 평균값을 계산 한 후 다음 프레임에서 객체를 그림 8과 같이 8방향으로 탐색 한 후 T1+3△처럼 탐색된 객체의 위치로 CAM shift의 중심점을 옮기고, 탐색 윈도우를 마지막 추적시의 탐색 윈도우의 크기로 설정 한 후 다시 CAM shift를 이용하여 추적을 시작한다.

Input	$Centers_{n-1}$
Process	Calculate $v_0 \sim v_8$ $v_0 = 8-way window(Centers_{n-1})$ $v_{1\sim 8} = 8-way window(Centers_{n-1} \pm 8way shift)$ $Centers_n = find(abs(\min(v_0 - v_{1\sim 8})))$
Output	$Move CAMwindow(Centers_n)$ $Resize CAMwindow()$

그림 12. 방향 탐색 윈도우의 객체 재 추적 의사코드

그림 12은 방향 탐색 윈도우의 객체 재 추적 의사코드이다. 일반적으로 지역 최대 점에 들어가거나 갑작스레 추적하는 객체를 놓치게 되었을 경우 CAM shift의 탐색 윈도우는 줄어드는 것을 확인할 수 있었고, 임계값으로 약 이전의 윈도우 사이즈보다 60%미만으로 줄어들었을 경우 추적을 실패 했다고 상정 하였다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

객체를 놓치게 되면 8방향 탐색 윈도우를 이용한 추적이 시작되는데, 이전에 마지막으로 추적에 성공한 위치인 $Center_{n-1}$에서 계산된 8방향 탐색 윈도우 값을 v_0라고하고 각각 8방향의 탐색 윈도우 값을 $v_1 \sim v_8$이라 할 때 v_0와 $v_1 \sim v_8$각각의 값의 차의 절대값을 구한 후 그 차가 가장 작은 곳으로 이동하여 CAM shift의 탐색 윈도우의 크기를 실패전의 크기로 조절하여 추적을 재개 한다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

제 4장 실험 결과

제 1절 실험 및 결과 분석

제안하는 알고리즘과 기존의 알고리즘을 동일한 영상의 객체를 추적하게 하여 각각의 성능을 비교 하였다.

제안하는 알고리즘은 기존 CAM shift의 프레임의 추적 간격을 2프레임 간격으로 늘리고 추적하는 대상의 급격한 이동을 통해 대상을 놓치게 되었을 경우 8방향 탐색 윈도우를 이용하여 8방향 탐색을 실시해 CAM shift가 객체를 재 추적 할 수 있게 하는 것이다.

표 1. CAM shift의 프레임별 연산량과 추적율

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

위의 표 1은 제안하는 알고리즘에서 추적하는 단위의 프레임 간격을 조절 하였을 때 연산량의 감소량과 추적율을 나타낸 것이다. 추적율은 각기 다른 추적 객체를 추적하기 위해 각기 다른 속도에서 각기 다른 움직임을 보이는 10개의 영상에 대한 추적 성공률이다.

추적율 실험에 사용한 영상은 서로 다른 속도로 이동하는 객체에 대한 추적이기 때문에 프레임 간격이 늘어나게 된 경우 느린 속도로 이동하는 객체에 대해서는 추적이 가능했지만 일반적인 속도로 이동하는 객체에 대해서는 추적을 하지 못하는 모습을 관측하게 되었고, 연산량과 추적율을 고려하여, 최적의 효율이라고 생각되는 2프레임 간격을 이용하여 실험을 진행 하였다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

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 14은 일반적인 CAM shift의 객체 추적이다. 매 프레임 마다 추적을 시행하는 동안 찍힌 중심점을 그래프로 나타내었다.

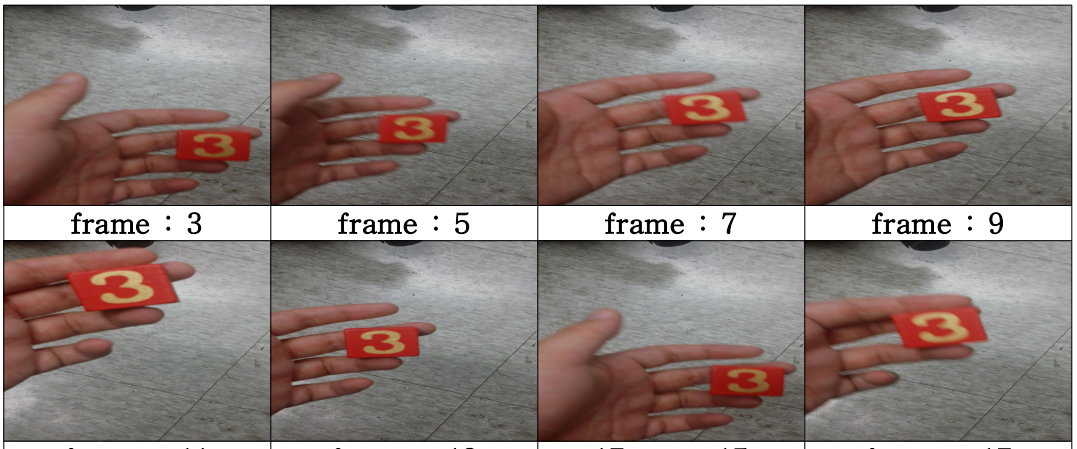
그림 15는 제안하는 방법대로 2프레임 간격으로 CAM shift의 추적 간격을 늘려 객체를 추적하였다.

그림 14과 15를 통해서 CAM shift의 추적 범위를 멀리 벗어나지 않는 객체에 대한 추적은 문제가 없음을 확인 할 수 있었다.

이어서 대부분의 영상에서도 2프레임 간격의 CAM shift를 이용해서 추적이 가능한지 테스트를 시행하여 보았고, 이것은 표 1의 추적율이다. 서로 다른 10개의 영상에 대해서 일반적인 CAM shfit는 9번의 추적성공을 보였고 제안하는 2프레임 간격의 추적은 8번의 성공을 확인 할 수 있었다.

2프레임 간격에서 CAM shift의 탐색 윈도우 영역 이상으로 이동을 하게 되었을 경우에는 CAM shift의 문제점인 다음 프레임에서 탐색 윈도우 영역 밖으로 추적 객체가 나간 것과 같은 현상이 벌어지게 되었다.

하지만 제안하는 알고리즘의 8방향 탐색 윈도우의 탐색 영역 안에 객체가 있어 추적을 재개 하는 모습을 확인 할 수 있었다.



frame : 3	frame : 5	frame : 7	frame : 9
frame : 11	frame : 13	17ame : 15	frame : 17

그림 13. 실험 영상(1)

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

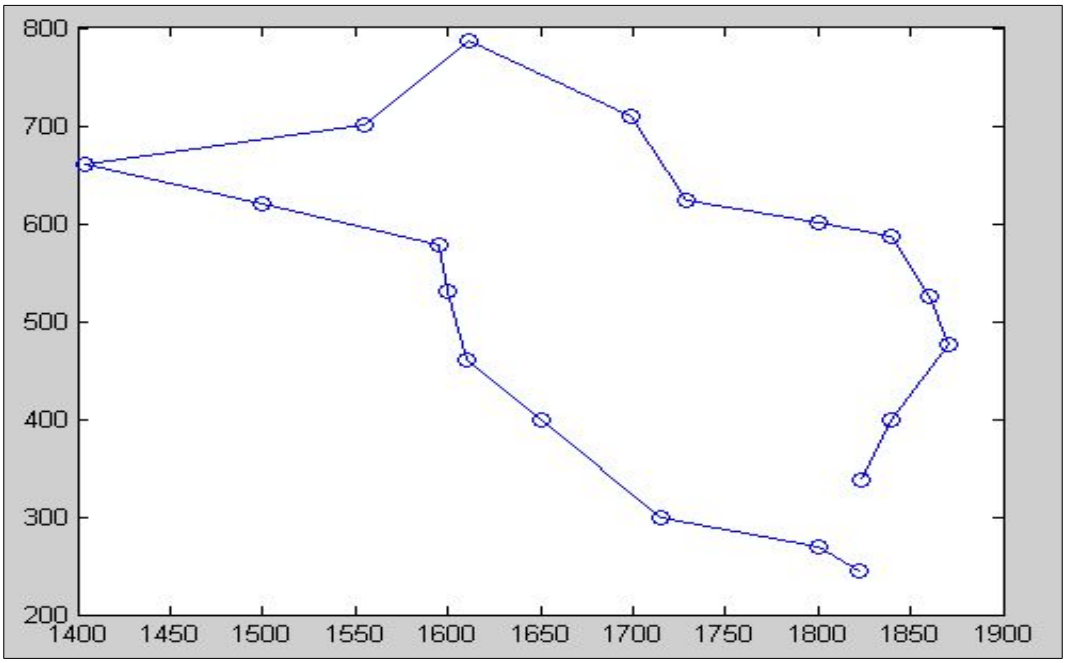


그림 14. 일반적인 CAM shift의 중심점

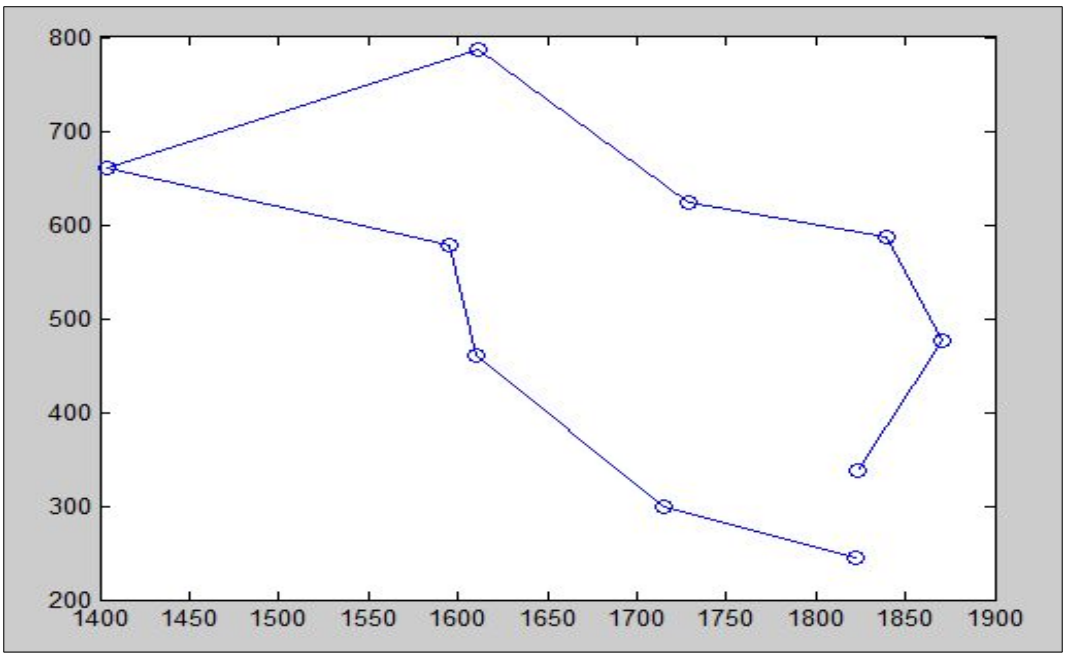


그림 15. 2프레임 간격의 CAM shfit의 중심점의 좌표

Collection @ chosun

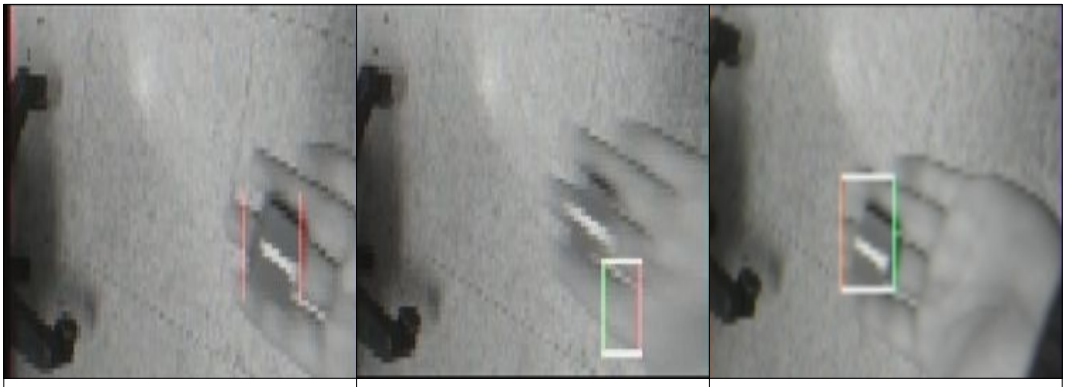
조선대학교
CHOSUN UNIVERSITY

8방향 탐색 윈도우를 적용시킨 CAM shfit를 사용하기에 앞서 일반적인 CAM shift를 이용하여 객체를 추적하였고, 고속 이동을 하는 10프레임이후 객체를 추적하지 못하는 모습을 확인 할 수 있다.

이 영상에 대해 8방향 탐색 윈도우를 적용시킨 CAM shift를 사용해 보았고, 다음 그림 16와 같은 결과를 얻을 수 있다.

8프레임까지는 문제없이 추적을 하는 모습을 확인할 수 있었고, 객체의 이동속도가 빨라진 10프레임에서 객체를 놓치게 되고 객체가 존재하지 않는 곳으로 탐색 윈도우가 이동한 모습을 확인 할 수 있다.

CAM shift가 객체를 놓친 상태임이 확인되자 8방향 탐색 윈도우에 의해 마지막 추적 성공위치에서 8방향 탐색 윈도우의 추적을 개시 하였고, 12프레임에서 CAM shift의 추적 윈도우가 8프레임의 크기로 재설정되어 추적을 재개한 모습을 확인 할 수 있었다.



frame : 8	frame : 10	frame : 12

그림 16. 고속 이동 객체 추적

고속으로 이동하는 객체를 놓치게 된 경우 중 제안하는 알고리즘으로 추적 가능한 영상과 그렇지 않은 영상이 있다.

표 2는 CAM shift와 8방향 탐색 윈도우가 적용된 CAM shift의 고속이동 객체 추적의 성공률이다. 서로 다른 10개의 영상에 대해 CAM shift는 5개의 성공을 보였고, 8방향 탐색 윈도우는 8개의 성공을 보였다.

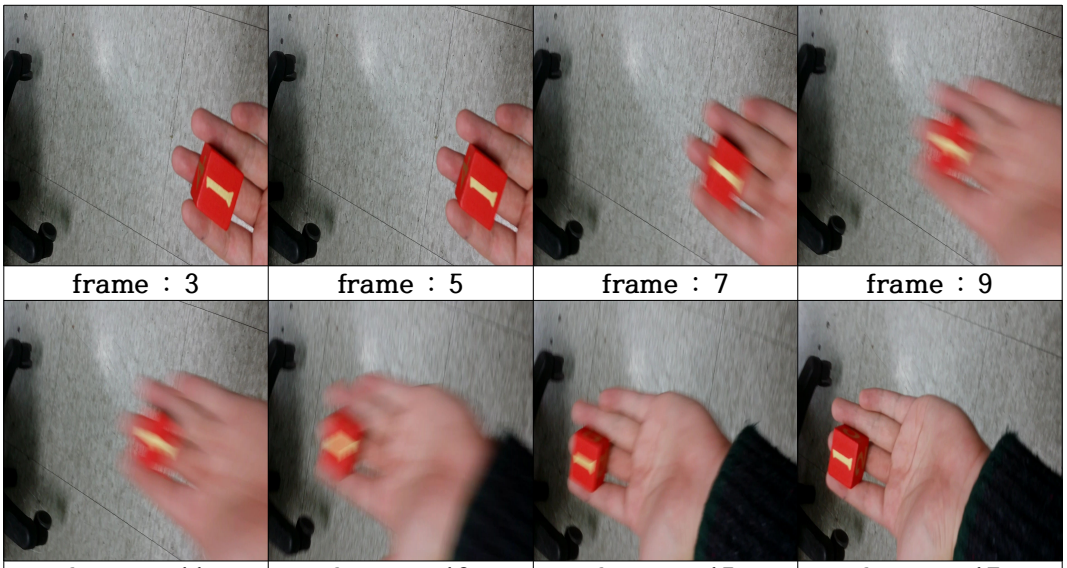
8방향 탐색 윈도우의 탐색 영역을 넘어서 이동해 버린 객체에 대해서는 8방향 탐색 윈도우를 이용해도 추적 할 수가 없었다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

표 2. 8방향 탐색 윈도우가 적용된 CAM shift의 고속 이동 객체 추적 성공율

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



frame : 3	frame : 5	frame : 7	frame : 9
frame : 11	frame : 13	frame : 15	frame : 17

그림 17. 실험 영상(2)

그림 17은 그림 18에 사용된 실험영상의 프레임이다. 정상적으로 추적을 하였을 경우 2프레임 간격인 3, 5, 7, 9, 11, 13, 15, 17 프레임을 추적하게 되며 추적한 프레임내의 객체의 색상의 중심을 따라 CAM shift가 추적을 지속하게 되나, 고속이동을 하여 CAM shift의 탐색 윈도우 밖으로 추적 객체가 나가게 된 경우에는 8방향 탐색 윈도우를 이용하여 객체를 찾고 CAM shift의 탐색 윈도우와 탐색창의 크기를 조절하여 재추적을 실시한다. 그림17의 영상의 크기는 1920 * 1080 이며 MATLAB2010A에서 실험을 실시하였다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

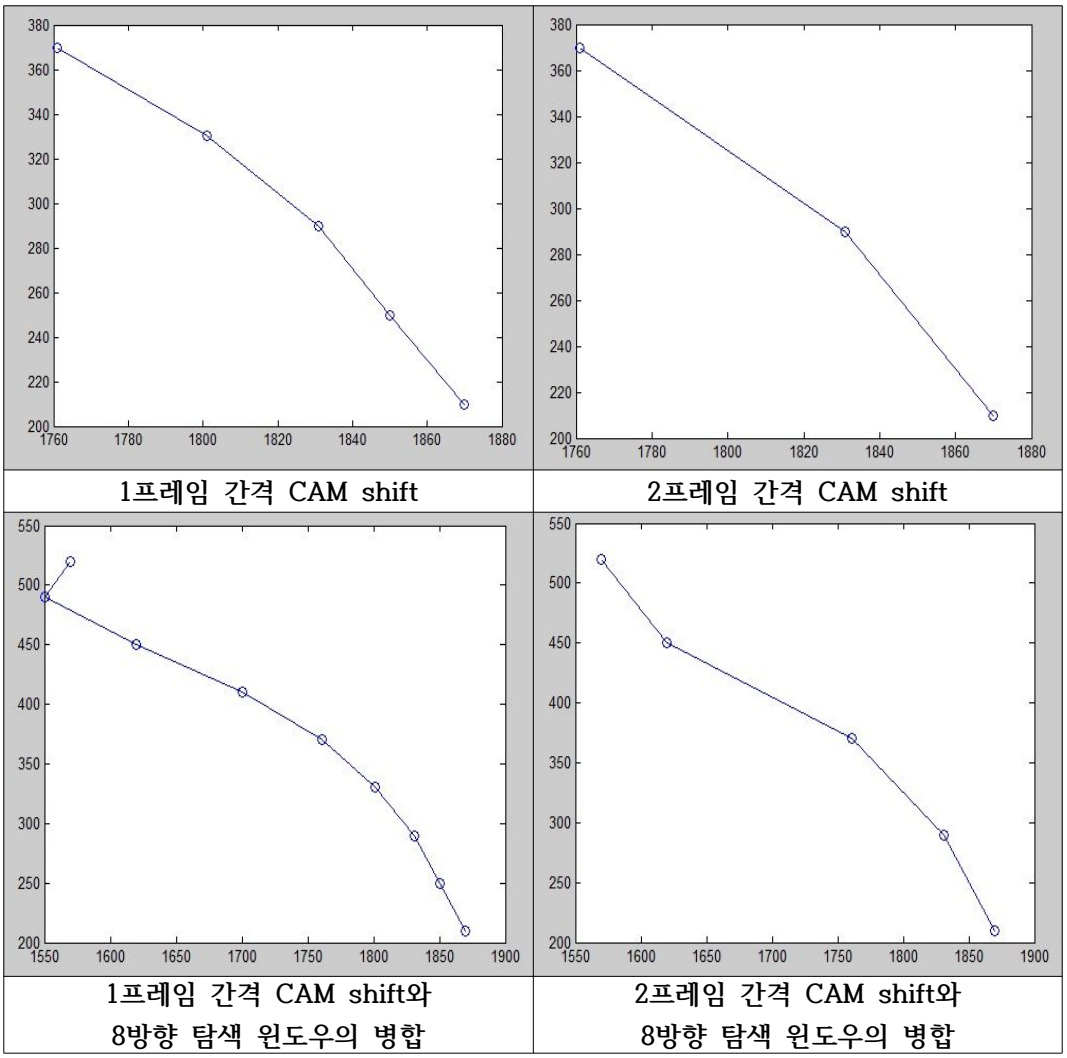


그림 18. 일반적인 CAM shift와 8방향 탐색 윈도우의 병합

그림 18은 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 shift

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

에 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.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

제 5장 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. 이에 따라 놓쳐버린 객체에 대해 재 추적이 가능하도록 하기 위해 CAM shift 알고리즘과 8방향 탐색 윈도우 병합한 방법을 제안하여 이를 실험 영상에 적용하였다.

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.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

참 고 문 헌

[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. 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

[3] 김 영균, “GP 기반 컬러 모델 생성과 개선된 CAMshift 알고리즘을 통한 이동 객체 추적에 관한 연구”, 서경대학교 대학원, 석사학위 논문, 2011

[4] 신진섭, 강대성 “Depth정보와 SURF 알고리즘을 이용한 실시간 객체 추적 알고리즘 연구,“ 2012 한국 정보 기술 학회 하계 학술대회 논문집, pp. 144-148 2012

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

[6] 박종대, 강대성 “배경 모델링과 Modified CAMshift를 이용한 실시간 상황 감지 시스템,” 한국 정보기술학회 논문지, 제 9권, 제 3호, pp. 207-212, 2011. 3.

[7] Kalman, R.E. “Anew approach to linear filtering and prediction problems”. Journal of basic Engineering, Vol. 82, No. 1, pp. 35 ~ 45, 1960

[8] 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

[9] 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

[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] 김대영, 박재완, 이칠우 “CAMshift 기법과 칼만 필터를 결합한 객체 추적 시스템,” Journal of Korea Multimedia Society Vol. 16, No. 5, 2013. 5, pp. 619-628

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[12] 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

[13] Zhou Bin, Wang Jun-zheng, Li Jing, and Shen Wei, "Object Tracking with Mean shift and Model Prediction," International congress on Image and Signal Processing, pp. 1-5, 2009

[14] 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

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

Collection @ chosun