

적응적 3 프레임 차분 방법 기반 템플릿을 이용한 객체 추적

Object Tracking Using Template Based on Adaptive 3-Frame Difference

김헌기*, 이진형*, 조성원*, 정선태**, 김재민*

Hunki Kim, Jinhyung Lee, Seongwon Cho, Sun-Tae Chung, Jaemin Kim

* 홍익대학교 전기정보제어공학과

** 송실대학교 정보통신전자공학부

요 약

물체를 추적하는데 있어서 추적하고자 하는 물체를 검출하여 템플릿을 만드는 것과 두 물체가 겹쳐지거나 다른 배경에 가려진 물체를 구분하여 추적하는 것은 물체 추적에 있어서 중요한 문제이다. 물체를 검출하여 템플릿을 만드는 방법으로 Frame Difference를 이용하면 천천히 움직이는 물체를 잘 구분할 수 없는 문제점이 있다. 이를 해결하기 위하여 본 논문에서는 Adaptive 3-Frame Difference를 이용하여 정확한 물체의 템플릿을 생성하는 알고리즘을 제안한다.

Abstract

To generate the template of a detected object and to track the overlapped object and the object covered by other objects correctly are important research problems in visual surveillance. The frame difference is not capable of generating the template of slowly moving object. To get around the drawback of the conventional frame difference, we propose a new algorithm for generating a template using adaptive 3-frame difference.

Key Words : object tracking, adaptive 3-frame difference, background subtraction, template

1. 서 론

최근 영상을 이용한 물체 추적은 컴퓨터 비전 분야에서 활발하게 연구되는 주제 중 하나이다. 물체 추적의 과정을 크게 구분하면 움직임 검출, 검출된 물체 분류, 물체 추적, 물체의 행동 분석, 다중 채널 카메라에서 들어오는 영상 처리 결과 결합 등으로 나눌 수 있다. 그림1은 이러한 영상 감시의 흐름을 나타내고 있다 [1].

물체 추적에 있어서 가장 큰 문제점은 두 물체가 겹치거나 겹침 후 분리될 때, 또는 물체가 배경에 가려져 사라진 후에 다시 나타나게 됐을 때 물체를 정확히 판별하지 못하는 것이다. 이를 해결하기 위하여 템플릿 매칭을 사용해야 하는데 단순한 배경 분리 (Background Subtraction) 방법을 이용하면 조도의 변화에 민감하기 때문에 본 논문에서는 Adaptive 3-Frame Difference를 이용하여 템플릿을 생성하고 그것을 이용하여 객체를 판별하는 알고리즘을 제안한다.

2. 움직이는 객체 검출

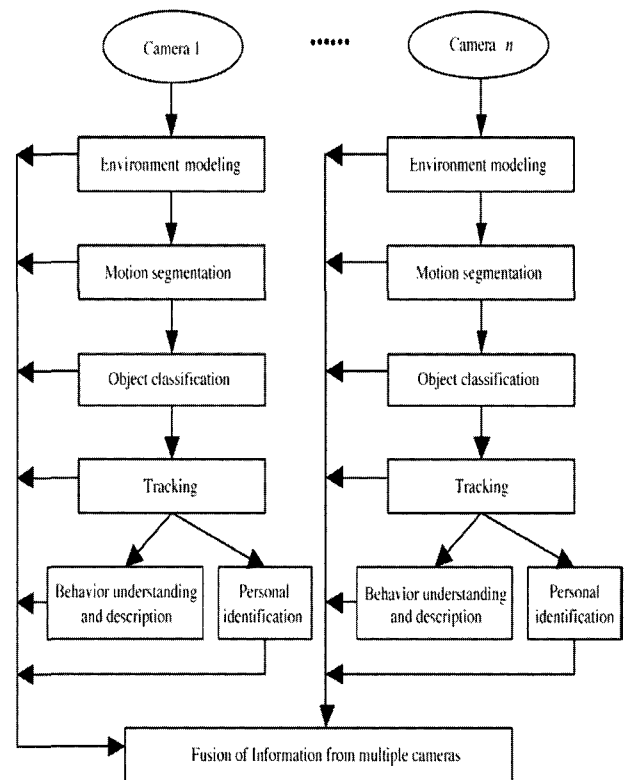


그림 1. 일반적인 영상 추적의 구성도.

Fig. 1. General Framework of visual surveillance.

접수일자 : 2007년 4월 1일

완료일자 : 2007년 5월 29일

감사의 글 : 본 연구는 2006년 산학협동재단의 지원을 받아 수행된 연구임.

움직이는 물체를 검출하는 방법은 크게 배경분리 방법과 프레임간의 차이를 이용하는 방법(Frame Difference)이 있다 [2]. 두 방법은 각각 문제점이 있는데 그림 2에서 보는 것과 같이 배경 추출 방법은 배경으로 분리되어 있던 물체가 움직일 시 빈 부분이 생겨 움직이는 물체로 오인되는 문제가 있고 프레임간의 차이를 이용하는 방법은 명암의 차이가 주로 외각에 분포되어 있어서 중심 부분이 잘 검출되지 않는 문제가 있다. 이를 해결하기 위하여 두 방법을 결합한 방법을 사용해야 한다 [3].

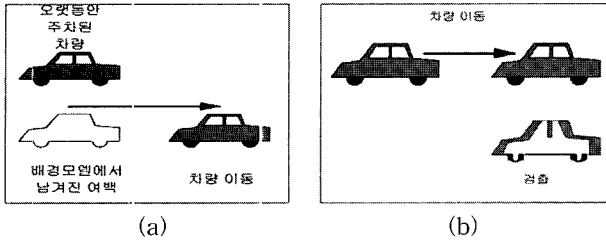


그림 2. 움직임 추출 방법의 문제점.

- (a) 배경 분리 방법의 문제점.
- (b) 프레임 간 차이를 이용한 방법의 문제점.

Fig. 2. The problem of detecting moving object.

- (a) The problem of background subtraction.
- (b) The problem of frame difference.

$$M_n(x) = (|I_n(x) - I_{n-1}(x)| > T_1 \text{ and } (|I_n(x) - B_n(x)| > T_2) \quad (1)$$

- $M_n(x)$: 움직임이 있는 영역
- $I_n(x)$: 현재 프레임의 화소값
- $I_{n-1}(x)$: 과거 1 프레임 전의 화소값
- $B_n(x)$: 배경 프레임의 화소값
- T_1 : Frame Difference의 임계값
- T_2 : 배경 분리 방법의 임계값

T_1 과 T_2 는 각각 방법이 따른 임계치를 나타내고 있다. 두 방법을 결합하여 물체를 추출하면 움직이는 부분만 프레임 차이를 이용하여 검출한 후 배경 분리 방법으로 정확한 물체 모양을

검출할 수 있다. 프레임 차이를 이용하는 방법 이용 시 중심 부분을 채우기 위하여 그림 3과 같이 침식과 팽창을 이용하여 넓은 지역을 검출해야 한다.

3. 템플릿 생성

지역 정보 기반(Region Tracker) 물체 추적은 움직이는 물체를 검출하고 검출된 객체가 새로운 객체라고 판별되면 새로 번호를 부여하고 이전 프레임에서 예측된 객체일 경우 계속적으로 추적하는 방법이다 [4], [5]. 그림 4는 지역 기반 추적을 이용한 물체 추적의 예를 보여주고 있다.

이러한 지역 정보 기반의 물체 추적은 두 물체가 가까이 있거나 겹침 현상이 있는 경우 추적중인 객체를 혼동할 우려가 있다. 이를 대비하여 각각 객체의 템플릿을 생성하여 겹침 현상이 발생하거나 겹침 후 분리될 때 계

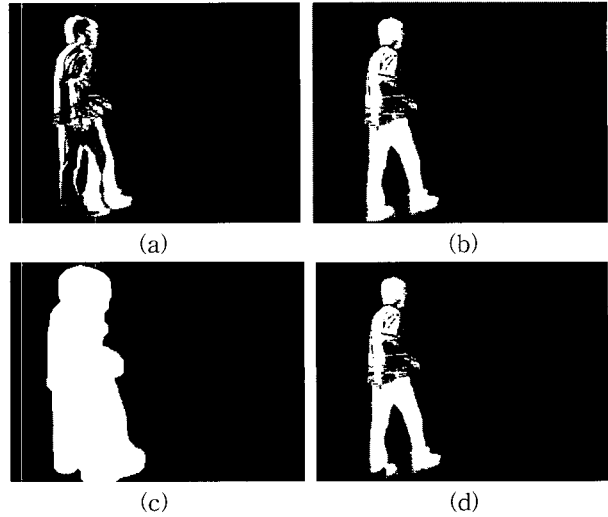


그림 3. 물체 추출 과정.

- (a) 프레임 차이를 이용한 물체 추출.
- (b) 배경 분리 방법을 이용한 물체 추출.
- (c) 프레임 차이 이후 팽창 및 침식을 이용한 물체 추출.
- (d) (b), (c) 두 방법을 결합한 방법을 이용한 물체 추출.

Fig. 3. The process of detecting object.

- (a) Object detection by frame difference.
- (b) Object detection by background subtraction.
- (c) Object detection by morphology technique after frame difference.
- (d) Object detection by fusion of two methods (b), (c).

속적으로 객체를 추적할 수 있도록 대비해야 한다.

템플릿을 생성하는 방법도 마찬가지로 배경 분리 방법과 프레임 차이를 이용하는 방법을 사용할 수 있는데 배경 분리 방법은 물체의 윤곽을 잘 추출할 수 있으나 배경의 조도가 변할 시 문제가 발생할 수 있고 프레임 차이를 이용하는 방법은 객체가 천천히 이동할 시 잘 검출하지 못하는 문제점이 있다. 이 문제점을 해결하기 위하여 Adaptive 3-frame difference를 이용한 방법을 사용할 수 있다.

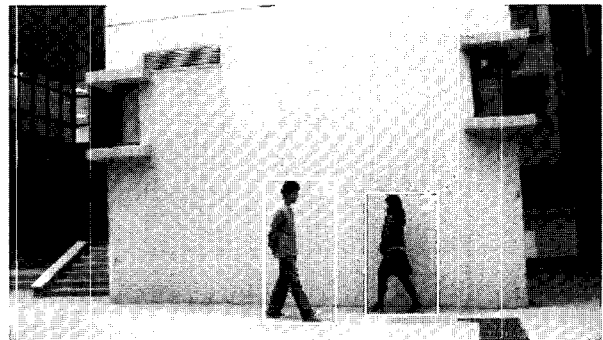


그림 4. 지역 기반 추적.

Fig. 4. Tracking based on Region tracker.

4. 제안하는 알고리즘

4.1 3-Frame Difference

프레임 간의 차이를 이용하여 템플릿을 만들 경우 두 개

의 프레임을 이용하여 템플릿을 생성하면 전 프레임과 현재 프레임의 객체가 모두 포함되어 정확한 모양의 템플릿을 만들 수가 없다. 이를 해결하기 위하여 3개의 프레임의 차이를 이용하여 템플릿을 생성하면 정확한 모양의 템플릿을 만들 수 있다. 그림 5에서 표현한 것과 같이 현재 프레임과 이전 2 프레임을 이용하면 완전한 모양의 템플릿을 만들 수 있다.

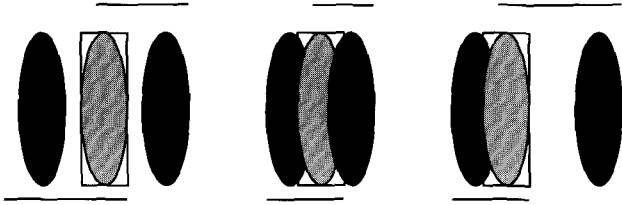


그림 5. 3-frame difference 도식화.

Fig. 5. The diagram of 3-frame difference.

$$R_n(x) = (|I_{n-2}(x) - I_{n-1}(x)| > T \text{ and } (|I_{n-1}(x) - I_n(x)| > T) \quad (2)$$

- $R_n(x)$: 템플릿으로 생성된 영역.
- $I_n(x)$: 현재 프레임의 화소값.
- $I_{n-1}(x)$: 과거 1 프레임 전의 화소값.
- $I_{n-2}(x)$: 과거 2 프레임 전의 화소값.
- T : Frame Difference의 임계값.

식 (2)에서 표현한 것과 같이 2 프레임 전의 화소값과 과거 1 프레임 전의 화소값의 차이를 계산하여 임계치 이상이면 움직임이 있는 영역으로 분류하고, 마찬가지로 과거 1 프레임 전의 화소값과 현재 프레임의 화소값의 차이를 계산하여 움직이는 영역을 찾은 후 두 부분이 공통으로 들어가는 영역을 계산하여 템플릿을 생성할 수 있다. 실제로 템플릿이 만들어지는 시기는 n-1번째 프레임이므로 한 프레임 지연되는 것을 알 수 있다.

4.2 Adaptive 3-Frame Difference

4.1절에서 설명한 3-Frame Difference 방법을 사용하면 정확한 객체의 템플릿을 생성할 수 있지만 프레임 간의 차이를 이용하는 물체 검출의 가장 치명적인 단점인 천천히 움직이는 객체의 경우는 정확히 검출할 수 없는 문제가 있다. 그러므로 이를 해결하기 위하여 연속된 3 프레임이 아닌 완전한 형태의 템플릿이 생성될 수 있는 3 프레임을 이용하여 프레임간의 차이를 이용하는 방법을 사용하여 정확한 템플릿을 생성해야 한다. 그림 6은 이를 도식화하여 보여주고 있다.

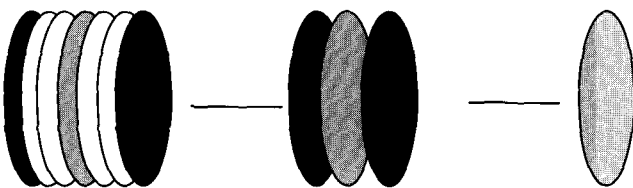


그림 6. Adaptive frame difference의 도식화

Fig. 6. The diagram of adaptive frame difference

$$R_n(x) = (|I_{n-l}(x) - I_{\frac{2n-l}{2}}(x)| > T \text{ and } (|I_{\frac{2n-l}{2}}(x) - I_n(x)| > T) \quad (3)$$

- $R_n(x)$: 템플릿으로 생성된 영역.
- $I_n(x)$: 현재 프레임의 화소값.
- $I_{n-l}(x)$: 과거 l 프레임 전의 화소값.
- $I_{\frac{2n-l}{2}}(x)$: 과거 l 프레임 전과 현재 프레임 간의 중간에 위치한 프레임의 화소값.
- T : Frame Difference의 임계값.

식 (3)에서 표현한 것과 같이 현재 프레임의 움직이는 영역의 화소값 $I_n(x)$ 과 이와 전혀 겹치지 않는 과거 l 프레임 전의 움직이는 영역의 화소값 $I_{n-l}(x)$ 을 찾고, 두 프레임 중간에 위치한 화소값 $I_{\frac{2n-l}{2}}(x)$ 을 4.1절에서 설명한 3-frame

difference 방법의 중간에 위치한 화소값으로 사용하면 완전한 형태의 템플릿을 생성할 수 있다. 이 방법을 사용하면 천천히 움직이는 물체뿐만 아니라 빠른 속도로 움직이는 물체도 마찬가지로 겹치지 않는 프레임을 찾고 프레임간의 차이를 구하기 때문에 정확한 템플릿을 생성할 수 있다.

2 프레임 또는 3 프레임의 차이를 이용하여 템플릿을 생성하면 천천히 움직이는 물체에 대하여 정확한 모양의 템플릿을 만들 수 없고 단순 배경 분리 방법을 이용하여 템플릿을 생성하면 배경이 변하는 상황에서 정확히 템플릿을 생성할 수 없다. 그러나 Adaptive 3-frame difference를 사용하면 겹치지 않는 3개의 프레임을 사용하기 때문에 연속적인 2개 또는 3개의 프레임의 차이를 이용한 방법의 문제점을 해결할 수 있고, 기본적으로 프레임간의 차이를 이용하는 방법이므로 프레임간의 시간적 차이가 배경 분리 방법에 비해 매우 짧기 때문에 조도의 변화에도 민감하지 않다.

5. 템플릿 매칭

두 객체간의 겹침 현상이 발생할 경우 단순 지역 정보만 이용하여 객체를 판단하면 두 객체를 하나로 인식하거나 새로운 물체로 판단할 수 있다. 그러므로 겹쳐진 상황에서 두 객체를 정확히 판단하기 위하여 본 논문에서 제안한 알고리즘을 이용하여 생성한 템플릿을 사용하여 템플릿 매칭 알고리즘(Template Matching Algorithm)으로 각각의 객체를 분류해야 한다. 하지만 템플릿 매칭만으로는 정확한 판별이 어려우므로 색 정보를 이용하여 판단하는 방법을 병행하여야 한다[6]. 그림 7은 제안한 알고리즘으로 템플릿을 구성한 것을 보여주고 있다.

6. 실험

실험은 펜티엄 3.0GHz PC에서 실행하였다. 영상은 기존 프레임 간의 차이를 이용하는 방법과 배경 분리 방법을 본 논문에서 제안하는 알고리즘을 사용한 방법과 비교하기 위하여, 배경의 외형적 변화가 거의 없는 곳에서 느리게 이동하는 두 객체가 겹쳐지는 상황, 배경의 조도 변화가 있는 상황을 설정하여 촬영하였다. 그리고 프레임 간의 차이를 이용한

는 방법과 배경 분리 방법에 사용한 임계값은 40으로 동일하게 설정하였다.

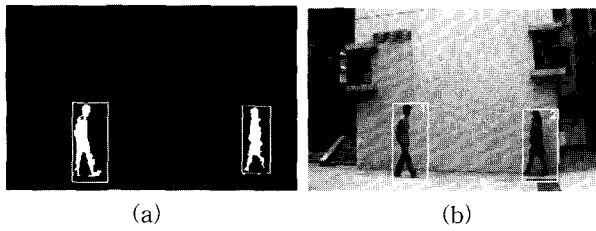
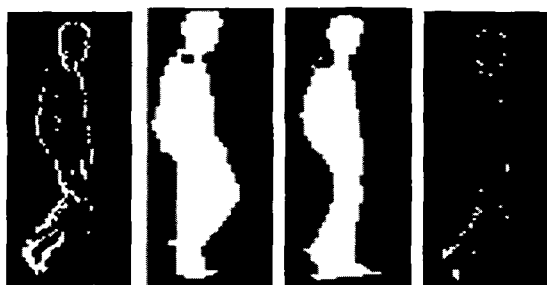
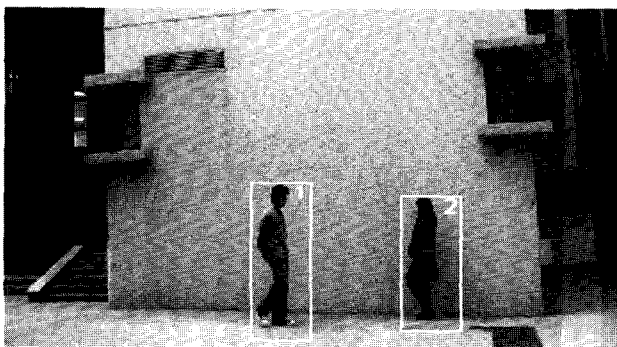


그림 7. 제안한 알고리즘으로 템플릿을 생성.
(a) 객체 추적 상황. (b) 객체의 템플릿 표현.

Fig. 7. Template generated by the proposed algorithm.
(a) Object tracking.
(b) The template generated by proposed algorithm.

6.1 느린 속도의 객체에 대한 템플릿 생성

느린 속도로 움직이는 사람의 움직임을 추적하는 상황에서 1번 객체에 대하여 각각 다른 방법으로 템플릿을 생성한 것을 비교해 보면, 2-Frame Difference를 이용한 방법과 3-Frame Difference를 이용한 방법은 객체의 움직임이 느리기 때문에 프레임 간의 차이가 거의 없어서 움직임 영역의 중심 부분 및 외형부분이 잘 검출되지 않고 그로 인하여 완전한 형상의 템플릿을 만들 수 없다.



(b) (c) (d) (e)

그림 8. 느린 속도로 움직이는 객체에 대한 각각 다른 방법을 사용한 템플릿 생성.

(a) 객체 추적 상황. (b) 2-Frame Difference 방법.
(c) 3-Frame Difference 방법. (d) 배경 분리 방법.
(e) 제안한 알고리즘을 사용한 방법.

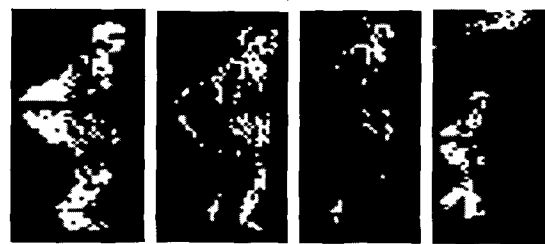
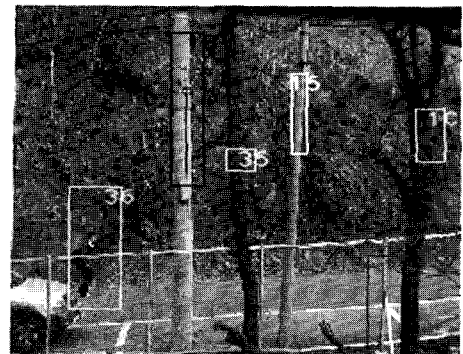
Fig. 8. Generation of a template for slowly moving object using several methods.

(a) Object tracking.
(b) Template by 2-frame difference.
(c) Template by 3-frame difference.
(d) Template by background subtraction.
(e) Template by the proposed algorithm.

그러나 배경 분리 방법이나 제안한 알고리즘을 이용한 방법은 완전한 형태의 템플릿을 만드는 것을 볼 수 있다. 실험에서 알 수 있듯이 조명 변화가 없는 상황에서는 제안한 알고리즘을 사용한 방법과 배경 분리 방법의 차이가 없는 것을 알 수 있다. 그림 7은 각각 다른 방법으로 1번 객체의 템플릿을 생성한 것을 비교한 것이다.

6.2 조명 변화 환경에서의 템플릿 생성

6.1절의 결과를 살펴보면 배경의 조명 변화가 없는 상황에서는 배경 분리 방법과 제안한 알고리즘의 차이가 거의 없다. 하지만 배경의 조명이 변하는 상황에서는 제안한 알고리즘이 배경 분리 방법보다 더 나은 성능을 보여주는 것을 확인할 수 있다. 그림 8은 조명 변화가 있는 상황에서 36번 객체의 템플릿을 생성한 것을 비교한 것이다. 본 영상은 같은 지역에서 햇빛이 비추지던 부분이 구름으로 가려지면서 배경의 조명 변화가 일어난 상황을 촬영한 것이다. 배경의 조명



(b) (c) (d) (e)

그림 9. 조명 변화 환경에서의 각각 다른 방법을 사용한 템플릿 생성.

(a) 객체 추적 상황. (b) 2-Frame Difference 방법.
(c) 3-Frame Difference 방법. (d) 배경 분리 방법.
(e) 제안한 알고리즘을 사용한 방법.

Fig. 9. Generation of a template in the environment of variant luminance using several methods.

(a) Object tracking.
(b) Template by 2-frame difference.
(c) Template by 3-frame difference.
(d) Template by background subtraction.
(e) Template by the proposed algorithm.

변화가 일어난 상황에서도 2-Frame Difference 방법이나 3-Frame Difference 방법은 객체의 중심 부분이 잘 검출되지 않기 때문에 완전한 형상의 템플릿을 만들 수 없다. 그리고 배경 분리 방법의 경우 배경에서 햇빛이 비춰지던 부분이 구름에 가려진 상황에서 객체가 그 부분의 근처를 지나갈 때, 객체와 배경에서 조도가 변한 부분이 겹쳐져서 하나의 템플릿으로 합쳐졌기 때문에 정확한 템플릿을 생성하지 못한 결과를 확인할 수 있다. 그러나 본 논문에서 제안한 알고리즘의 경우 배경의 조도 변화가 일어난 상황에서도 정확히 템플릿을 생성한 것을 확인할 수 있다.

6. 결과 및 향후 계획

본 논문에서 Adaptive 3-Frame Difference 방법을 이용하여 움직이는 객체의 템플릿을 만드는 방법에 대하여 제안하였다. 객체가 느리게 움직이는 상황이나 배경의 조명 변화가 있는 상황에서 템플릿을 만드는 방법 중 본 논문에서 제안한 Adaptive 3-Frame Difference를 이용한 방법이 가장 좋은 결과를 보여주고 있는 것을 확인할 수 있다.

그러나, 조명 변화에 의하여 실제로 움직이는 물체가 아닌 경우에도 객체로 잡는 문제가 발생한 것을 확인할 수 있다. 그러므로 이러한 문제점을 해결하기 위하여 배경을 업데이트 하는 방법을 사용하여 실제로 움직이는 물체만 객체로 추출해야만 한다. 그리고 템플릿을 생성하기 위하여 겹치지 않는 다음 프레임을 찾는 과정에서 움직이던 물체가 겹치지 않는 다음 프레임을 찾기 전에 정지하는 경우 무한히 프레임을 저장하는 문제가 발생할 수 있는데 이는 일정 프레임 이하로 프레임을 저장하는 공간을 제안함으로써 해결할 수 있다. 그리고 움직임을 멈춘 물체는 레이어(layer)로 분류하여 배경과 함께 움직이지 않는 물체로 간주하는 방법을 사용해야 한다. 그리고 물체를 사람과 사람이 아닌 물체로 구분하여 사람으로 분류했을 경우 사람의 얼굴을 찾아서 확대한 후 얼굴 인식을 이용하는 방법을 사용할 수 있다.

참고 문헌

- [1] Weiming Hu, Tieniu Tan, Liang Wang, Maybank. S, "A survey on visual surveillance of object motion and behaviors", in Systems, Man and Cybernetics, Part C, IEEE Transactions on , 2004, pp. 334 - 352
- [2] C. Anderson, Peter Burt, and G. van der Wal. "Change detection and tracking using pyramid transformation techniques". In Proceedings of SPIE - Intelligent Robots and Computer Vision, volume 579, pp. 72-78, 1985.
- [3] Robert T. Collins, Alan J. Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt and Lambert Wixson, "A System for Video Surveillance and Monitoring", tech. report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May, 2000
- [4] Nils T Siebel, "Design and implementation of people tracking algorithms for visual surveillance

- applications", Reading Univ. March 2003
- [5] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4 : Real-time surveillance of people and their activities," IEEE Trans. Pattern Anal. Machine.
 - [6] Gary R. Bradski "Computer Vision Face Tracking For Use in a Perceptual User Interface", Intel Technology Journal, 1998, Microcomputer Research Lab, Santa Clara, CA, Intel Corporation.

저자 소개



김현기(Hunki Kim)

2005년 : 홍익대 전자전기공학부 졸업
2006년~현재 : 동 대학원 전기정보제어공학 석사 과정

관심분야 : 패턴 인식, 영상 처리, 신경회로망
Phone : 02 - 320 - 1493
Fax : 02 - 320 - 1110
E-mail : acehunki@hotmail.com



이진형(Jinhyung Lee)

2005년 : 홍익대 전자전기공학부 졸업
2006년~현재 : 동 대학원 전기정보제어공학 석사 과정

관심분야 : 패턴 인식, 영상 처리, 영상 추적 시스템
Phone : 02 - 320 - 1493
Fax : 02 - 320 - 1110
E-mail : pentera34@naver.com



조성원(Seongwon Cho)

1982년 : 서울대 전기공학과 졸업.
1992년 : Purdue Univ. 전기공학과 졸업 (공학)
현재 : 홍익대 전자전기공학부 교수

관심분야 : 퍼지 이론, 신경회로망
Phone : 02 - 320 - 1493
Fax : 02 - 320 - 1110
E-mail : swcho@wow.hongik.ac.kr



정선태(Sun-Tae Chung)

1983년 : 서울대학교 전자공학과 학사
1986년 : 미국 미시간대학교(앤아버) 전자
및 컴퓨터 석사
1990년 : 동 대학원 전자 및 컴퓨터 박사
1991년~현재 : 숭실대학교 정보통신전자
공 학부 교수

관심분야 : 패턴 인식, 얼굴 인식
Phone : 02 - 820 - 0638
Fax : 02 - 820 - 7653
E-mail : cst@ssu.ac.kr



김재민(Jaemin Kim)

1984년 : 서울대 전기공학과 졸업.
1994년 : R. P. I. 전기공학과 졸업(공학박).
현재 : 홍익대 전자전기공학부 부교수

관심분야 : 패턴 인식, 영상 처리
Phone : 02 - 320 - 1634
Fax : 02 - 320 - 1110
E-mail : jaemin@wow.hongik.ac.kr