

실시간 배경갱신 및 이를 이용한 객체추적

Real time Background Estimation and Object Tracking

이 완 주 (Lee, Wan Joo)*

목 차

- I. 서론
 - II. 배경생성 및 움직임 탐지
 - III. 객체추적
 - IV. 실험 및 결과
 - V. 결론
-

Abstract

Object tracking in a real time environment is one of challenging subjects in computer vision area during past couple of years. This paper proposes a method of object detection and tracking using adaptive background estimation in real time environment. To obtain a stable and adaptive background, we combine 3-frame differential method and running average single gaussian background model. Using this background model, we can successfully detect moving objects while minimizing false moving objects caused by noise. In the tracking phase, we propose a matching criteria where the weight of position and inner brightness distribution can be controlled by the size of objects. Also, we adopt a Kalman Filter to overcome the occlusion of tracked objects. By experiments, we can successfully detect and track objects in real time environment.

Key words: adaptive, background, track objects

* 정회원, 용인대학교 컴퓨터정보학과, 교신저자: 이완주(wjlee@yongin.ac.kr)

1. 서론

영상처리 기술 및 장비의 발달은 방대한 데이터의 처리를 요하는 동영상의 실시간 처리를 가능하도록 하는 수준에 이르렀다. 동영상의 처리는 크게 방대한 동영상 데이터를 효과적으로 압축/전송하기 위한 동영상 압축기술에 관한 연구와, 동영상으로 부터 3차원 객체모델의 복원, 비디오 감시 및 객체 추적, 또는 비디오 데이터의 효율적인 저장 검색을 위한 의미정보의 자동 추출 및 저장방법에 관한 연구 등의 분야로 나누어진다.

이러한 연구 중 실시간으로 동작하는 감시시스템에서 움직임 객체를 검출하여 추적하고, 더 나아가 행위인식에 관한 연구는 이러한 연구들이 가지는 광범위한 응용분야로 최근 활발한 연구가 진행되고 있다[1-7].

실시간으로 동작하는 감시 시스템은 크게 2가지 부류로 분류할 수 있는데, 첫번째는 카메라가 정지된 상태에서 실시간 감시 및 추적을 행하는 시스템이고 두번째는 팬틸트가 부착된 능동카메라를 이용한 시스템이다.

본 논문에서는 정지된 카메라로 부터 입력되는 실시간 영상으로 부터 감시 및 추적을 위한 방법을 제안하였다. 이러한 시스템에서 움직임의 검출은 크게 2가지로 분류할 수 있다. 첫 번째가 프레임차이를 이용하는 방법이고, 두 번째는 배경영상과의 차이를 이용하는 방법이다. [2,3,9]

프레임차이를 이용하는 방법의 경우, 이전 프레임과 현재 프레임의 차이를 통해 움직임 영역을 판단하는 방법으로 동적환경에서 효과적이지만, 물체의 완전한 외곽을 추출하기에는 어려움이 있다. 오늘날 지능적인 감시 시스템은 움직임의 탐지 뿐만 아니라, 움직임 물체를 배경과 분리하여 지속적으로 추적하는 것을 목표로 한다. 하지만 프레임 차이에 의해 얻어지는 움직임 영역은 실제 움직임 영역과 가리어져있다가 드러나는 배경영역, 이 두개의 영역을 모두 포함한다. 따라서 이 두개의 영역 중 어떤 영역이 실제 객체를 나타내는지 판단하기 위해서는 별도의 연산이 필요하고, 또한 움직임이 크지 않은 경우에는 정확하게 물체의 경계만을 추출하기에는 어려움이 따른다.

이를 개선하기 위해 세프레임 차이를 이용하는 방법도 있지만, 이도 여전히 움직임이 크지 않은 경우 물체의 경계를 얻어내기 힘들다[9].

최근의 추세는 움직이는 물체가 제거된 배경영상을 확보하고, 이를 현재 프레임과 비교하여 움직이는 물체를 탐지하는 방법이 주로 연구되고 있다[2,10]. 이 방법은 이론적으로 물체의 윤곽을 정확히 추출할 수 있는 장점이 있다. 이 방법이 효과적이기 위해서는 배경을 정확하게 얻어낼 수 있어야 하며 아울러 조명의 변화 등으로 시시각각 변화하는 배경을 적응적으로 모델링 할 수 있어야 한다.

본 논문에서는 세 프레임 차이에 의한 방법과 배경에 의한 방법을 결합하여 실시간으로 안

정된 배경모델을 형성하고 이를 이용하여 움직임 객체를 추출하여 추적하는 새로운 방법을 제시한다. II장에서는 움직임 물체의 탐지 및 적응적 배경모델 갱신에 대해 설명하고, III장에서는 탐지된 물체를 지속적으로 추적하기 위한 방법론을 제시한다. IV장에서 제안된 방법으로 실험한 결과를 제시하고, V장에서 결론을 내린다.

II. 배경생성 및 움직임 탐지

감시 시스템에서 우선적으로 해결해야 할 과제는 움직이는 물체를 효과적으로 추출해 내는 작업이다. 물체의 추출은 첫 번째, 최근 2프레임 또는 3프레임간의 시간차를 이용한 방법과 안정된 배경을 확보하고 배경과 현재프레임의 차이를 이용하는 방법, 그리고 광류(optical flow)를 이용하는 방법이 있다. 프레임차이를 이용하는 방법은 동적인 환경에서 효과적이지만 물체의 완전한 외곽을 추출하기에는 어려움이 있다. 반면 배경영상을 이용하는 방법은 완전한 물체의 외곽을 얻어낼 수 있지만, 조명 또는 외부환경의 변화 등의 요인으로 인해 안정된 배경확보가 어렵다는 단점이 있다. 광류를 이용하는 방법은 카메라의 움직임이 있는 경우에도 전체의 움직임과는 독립적인 움직임을 찾아낼 수 있어 효과적이지만 연산의 복잡하여 실시간 응용에는 어려움이 있다.

본 논문에서는 배경영상의 생성을 위해 조명 등 외부환경의 변화를 점진적인 변화로 보고 다음과 같이 이동 평균개념의 단일 가우시안 모델을 적용하였다. 즉, 현재의 배경영상을 B_n 과 V_n 을 다음과 같이 모델링 된다.

$$\begin{aligned} B_n &= (1 - \alpha)B_{n-1} + \alpha \overline{F_n}, \text{ where } B_0 = F_0 \\ V_n &= (1 - \alpha)V_{n-1} + \alpha \overline{D_n}, \text{ where } V_0 = \{th\} \end{aligned} \quad (1)$$

여기서 B_n 은 최근 α 프레임 동안의 이동평균으로 구성된 배경영상을 의미하고, $\overline{F_n}$ 은 현재 프레임 F_n 에서 움직임 영역이 제거된, 즉 움직임 영역이 이전 배경 B_{n-1} 로 채워진 영상을 의미한다. α 는 1과 0 사이의 상수값으로 배경의 수렴속도를 결정한다. 지나치게 작은 α 값은 배경의 적응성을 떨어뜨리고, 큰 α 값은 배경의 안정성을 떨어뜨린다. 실험결과 외부환경 이전 내부환경이건 α 값을 0.01 ~ 0.05 사이의 값을 적용하면 안정되고 외부환경변화에 적응해 나가는 좋은 배경을 얻을 수 있었다.

V_n 은 최근 α 프레임 동안의 이동평균 방식으로 구해진 각 픽셀의 표준편차를 나타낸다. 즉,

$\overline{D_n}$ 은 움직임 영역이 제거된 차영상으로, 위의 식으로 얻어진 V_n 은 배경영역 중 바람에 흔들리는 나뭇잎, 흔들리는 창틀 등의 실제 고려할 필요가 작은 성분의 움직임 영역에 대해서는 높은 값을 가지고, 안정된 배경에 대해서는 낮은 값으로 구성되어 움직임 여부를 판단할 때의 경계치로 사용하면 잡음에 강한 움직임 추출이 가능하다.

$\overline{F_n}$ 은 이전 프레임까지의 배경 B_{n-1} 이 확보된 경우 다음의 식 (2)로 간단하게 구할 수 있다.

$$\overline{F_n} = \text{Replace}(F_n, B_{n-1}, \text{FindObjs}((F_n - B_{n-1}))) \quad (2)$$

여기서 $\text{FindObjs}()$ 함수는 F_n 과 B_{n-1} 의 차영상으로부터 움직임 영역을 찾아내어, 그 영역을 둘러싸는 최소사각형의 좌표를 되돌려주는 함수이고, $\text{Replace}()$ 함수는 F_n 을 B_{n-1} 로 대체하는데, 마지막 인수로 정의된 사각형 영역부분만을 B_{n-1} 으로 대체시켜주는 함수이다.

$\text{FindObjs}()$ 함수의 동작은 다음과 같다.

Algorithm FindObjs

```

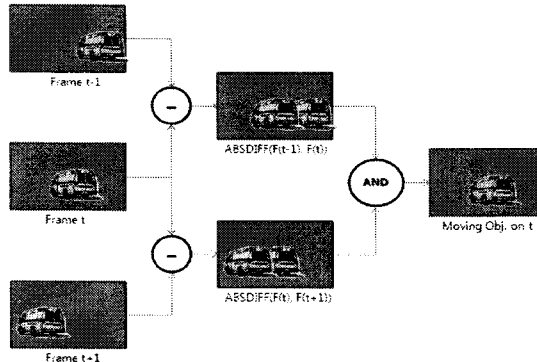
procedure FindObjs(F, B, V, R)
{ F is array of Image Data from Current Frame }
{ B is array of background Image and V is array of its deviation }
{ R is container of founded rectangles }
begin
  D := AbsDiff( F, B ): { D[i, j] = ABS(F[i, j] - B[i, j]); }
  D_bin := Threshold(D, MAX[2*V, min_th]): {Thresholding by 2*V }
  D_bin := Closing(D_bin, se): {se:rectangular structuring element }
  R := MER(D_bin): { Minimun Enclosing Rectangle of D_bin }
end

```

위의 알고리즘으로 실험을 진행했을 때 초기배경에 물체가 포함되지 않았고, 물체가 지속적인 움직임을 보이는 경우에 비교적 정확하게 움직임 물체를 탐지하였다. 하지만, 물체가 움직임을 보이다가 장시간 정지하는 경우(특히, 이 물체가 감시영역의 크기에 비해 매우 작은 경우)에는 배경과 객체의 경계부분에서 발생하는 약간의 밝기변화가 점진적으로 배경에 반영되어 일정시간이 지나면 배경에 물체가 완전히 포함되게 되어 이후부터는 물체가 추출되지 않는 단점이 드러났다. 아울러, 초기 배경의 생성 시 일정기간동안 움직임을 탐지하지 않고 지속적으로 현재 프레임을 배경에 누적시키게 되는데, 이 경우 배경 이미지에 움직임 물체가 이미 포함되어 있을 경우 이후 움직임 탐지에 어려움을 겪게 된다.

따라서 본 논문에서는 배경과 무관하게 적용할 수 있는 세 프레임 차이 방법을 사용하여 초

기배경을 모델링하고, 배경모델이 완전히 형성되고 나면 그 이후부터 배경차 방법을 이용하여 움직임을 탐지하고, 얻어진 움직임 정보를 이용하여 다시 배경을 갱신하는 방법을 사용했다. 세 프레임 차이에 의한 방법은 비록 1 프레임의 지연이 발생하지만 두 프레임차이에 의한 방법보다 정확하게 움직임 영역을 찾아낼 수 있다.



〈그림 1〉 3 프레임 차이에 의한 움직임 영역 추정

〈Fig. 1〉 Detecting Moving Object Region using 3 frame temporal differences

세 프레임차이 방법을 같이 적용한 경우, *FindObjs()*는 다음과 같이 변경된다.

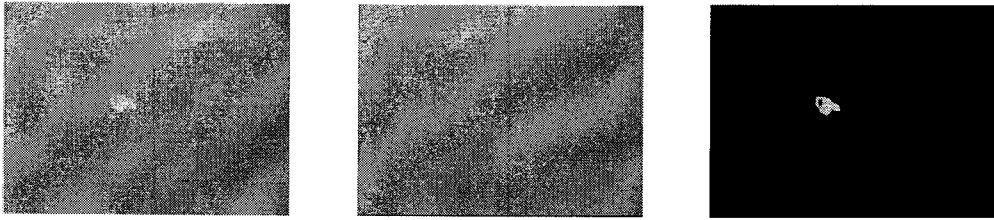
Algorithm FindObjs (modified)

```

procedure FindObjs(F2, F1, F0, B, V, R)
{ F2, F1, F0 are array of Image Datas from n+1th Frame, nth Frame, n-1th Frame }
{ B is array of background Image and V is array of its deviation }
{ R is container of founded rectangles }
begin
D1 := Threshold(AbsDiff(F2, F1), MAX[2*V, min_th]);
D2 := Threshold(AbsDiff(F1, F0), MAX[2*V, min_th]);
D_3f := Closing((D1 AND D2), se);
if(B is not yet prepared)
begin
R := MER(D_3f);
else { Modelling of Background already completed then use it. }
D1 := AbsDiff(F1,B): { D[i, j]: = ABS(F1[i, j] - B[i, j]) }
D2 := Threshold(D1, MAX[2*V, min_th]): {Thresholding by 2*V }
D_bg := Closing(D2, se);
R := { MER(D_bg) ∩ MER(D_3f) };
end
end
    
```

\overline{D}_n 은 움직임 물체를 구하는 과정에서 얻어지는 배경과 현재영상의 차영상을 의미한다. 이는 FindObjs()함수의 실행과정에서 부수적으로 얻어진다.

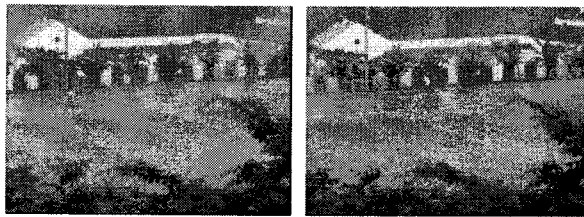
다음 그림은 위의 방법으로 얻어진 배경영상과 이를 이용한 움직임 물체 탐지의 결과를 표시한다.



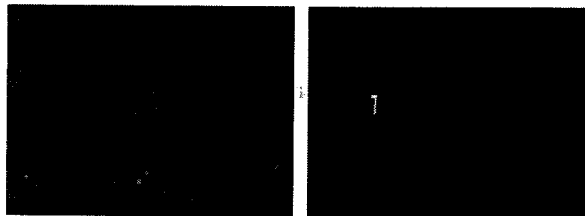
(a) 원영상 (b) 배경영상 (c) 움직임영역

〈그림 2〉 움직임 물체의 탐색 결과(실내)

〈Fig. 2〉 Results of Detecting Objects(Indoor)



(a) 원영상 (b) 배경영상(B_n)



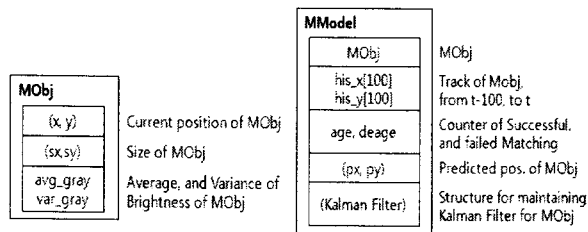
(b) 배경영상(V_n) (d) 움직임 영역

〈그림 3〉 움직임 물체의 탐색 결과(실외)

〈Fig. 3〉 Results of Detecting Objects(Outdoor)

III. 객체추적

II 장에서 얻어진 움직임 정보들은 크기, 위치, 밝기 등으로 구성된 특징벡터로 표현되고, 이들이 이후 프레임에서도 매칭에 성공하면 기본특징 외에 과거 움직임벡터의 위치이력, 이를 칼만 필터로 예측하여 얻어낸 객체의 다음 프레임에서의 위치정보, 매칭에 성공한 횟수 및 실패한 횟수등으로 구성된 객체모델로 변환, 유지된다 (그림 4) 움직임벡터(MObj) 및 객체모델(MModel))



〈그림 4〉 움직임벡터 및 객체모델

〈Fig. 4〉 Feature Vector of Moving Object and Model of Moving Objects

Algorithm Tracking

```

procedure Tracking(Detected, Moving, Temp)
{ Detected: list of Moving Objects just founded}
{ Moving: list of continuously matched Objects}
{ Temp: list of candidate Moving Objects}
{ initially Detected, Moving, Temp Lists are empty.}
begin
  if(Moving.count > 0) then
    begin
      DoMapping(Detected, Moving):
    end
  if(Temp.count > 0) then
    begin
      DoMapping(Detected, Temp):
    end
  Temp := Temp + Detected:
  CheckAge(Temp, 20):
end
    
```

FindObjs() 함수를 통해 얻어진 움직임벡터들은 일단 Detected 리스트에 보관된다. 현재까지

추적중인 객체들의 모델(MModel)은 각기 Moving, 또는 Temp 리스트에 보관된다. Moving 리스트는 지속적으로 매칭에 성공하고 있는 움직임 객체들에 대한 리스트이고, Temp 리스트는 현재 프레임에서 발견되었지만 매칭에 성공하지 못한 객체들을 다음 매칭에서 사용하기 위해 보관하는 리스트이다. Temp리스트의 객체들 중 지속적으로 매칭에 성공하는 객체들은 Moving 리스트로 옮겨진다.

객체모델의 age는 지금까지 객체가 매칭에 성공한 횟수를 나타내고, deage는 객체가 연속해서 몇번 매칭에 실패했는가를 기록하여, 움직임이 탐지된 경우라도 연속적으로 매칭에 실패할 경우에는 노이즈로 간주하여 삭제하도록 하였다.

움직임이 탐지된 개위의 알고리즘 추적의 성능을 결정짓는 주된 요소가 객체모델 및 유사도 측정 방법이다. 물체모델을 복잡한 모델을 사용하면 매칭의 정확도를 높일 수 있으나, 물체모델의 유지 및 갱신에 많은 비용이 요구되어 실시간 응용에는 아직까지 적합하지 못하다. 본 연구에서는 객체모델을 가장 기본적인 물체의 정보만을 이용하도록 하여 객체모델의 유지 및 갱신에 소요되는 비용을 최소화하였고, 아울러 정합방법으로는 아래식과 같이 가중거리의 개념을 도입하여 물체의 위치정보와 물체의 밝기정보를 물체의 크기에 따라 가변적으로 적용할 수 있도록 하였다.

즉, 물체의 크기가 작은 경우에는 그만큼 물체의 내부정보에 대한 중요도는 물체의 위치정보에 비해 낮아지고, 또, 물체의 크기가 감시영역에 비해 큰 경우에는 현재 물체의 위치보다는 물체의 내부정보가 더욱 중요한 역할을 하도록 하였다.

$$distance = (1 - \beta) \times eucd(MObj.pos, MModel.predicted) + \beta \times eucd(MObj.histo, MModel.histo) \quad (3)$$

여기서 $eucd()$ 는 유클리디안 거리를 말하며, β 는 물체의 크기에 따라 변하는 값으로 실험적으로

$$\beta = \text{Min}\left[\frac{4 \times \text{Object의 크기}}{\text{감시영역의 크기}}, 1\right] \quad (4)$$

로 정의하였다.

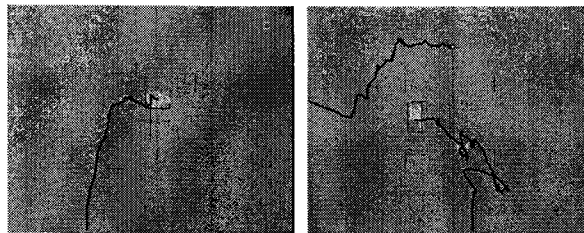
또한, 정합의 정확도를 높이고, 객체의 겹침, 가려짐 등에 효과적으로 대처하기 위해 정합 전에 과거위치정보와 현재의 위치정보로부터 다음 프레임에서의 위치정보를 선형칼만필터를 이용하여 예측하고, 예측된 위치정보를 정합에서 사용하도록 하였다. 이 경우 정합이 이루어지지 않더라도 프레임개수가 증가함에 따라 칼만필터에 의해 예측되는 위치정보도 지속적으로 갱신되므로 추적물체가 가려졌다가 다시 나타나는 경우에도 원래의 객체로 정합을 지속할 수 있음을 확인할 수 있었다.

IV. 실험 및 결과

실험 장비는 펜티움 4 기반의 PC와 컬러 CCD 카메라, 그리고 영상취득장치로 Matrox Meteor II 를 사용하였다. 실험에 사용된 프로그램은 Visual C++기반으로 OpenCV라는 공개된 비전라이브러리를 사용하였다.

취득된 영상은 640*480의 컬러 영상이지만, 이영상은 단순히 렌더링의 목적으로만 사용하고, 실제 처리에는 이 영상을 단일계조도 영상으로 변환한 후 320*240의 영상으로 축소하여 계산량을 줄이도록 하였다. 실제 객체의 추적 시에는 컬러정보가 매우 중요한 요소이지만, 본 논문의 최종 목표는 열 영상에 대한 물체추적장비였고, 열 영상은 단일계조도 영상과 동일한 성질을 가지므로 실험대상을 단일계조도 영상으로 하였다.

다음 그림은 실내환경에서 물체의 추적실험 결과를 보여준다. 그림 (a)는 단일물체의 추적결과를, (b)는 2개 물체의 추적결과를 나타낸다.



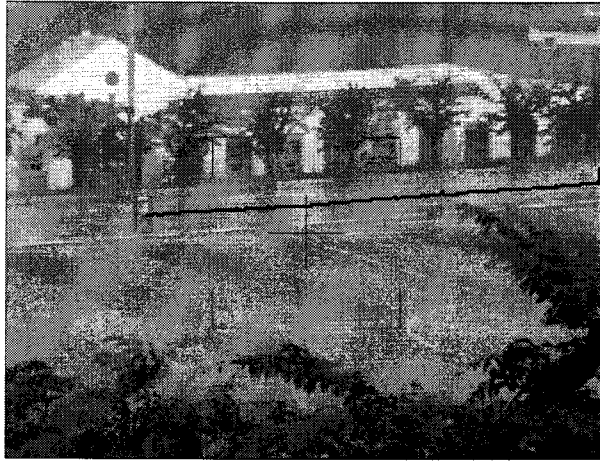
(a) 단일물체추적

(b) 2개물체추적

〈그림 5〉 실내환경의 물체추적

〈Fig. 5〉 Tracking object in indoor environment.

다음 그림은 실외환경에서의 물체추적 결과를 나타낸다.

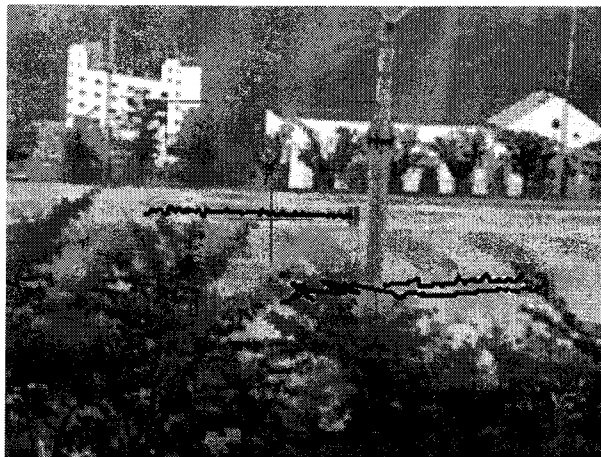


〈그림 6〉 실외환경의 물체추적

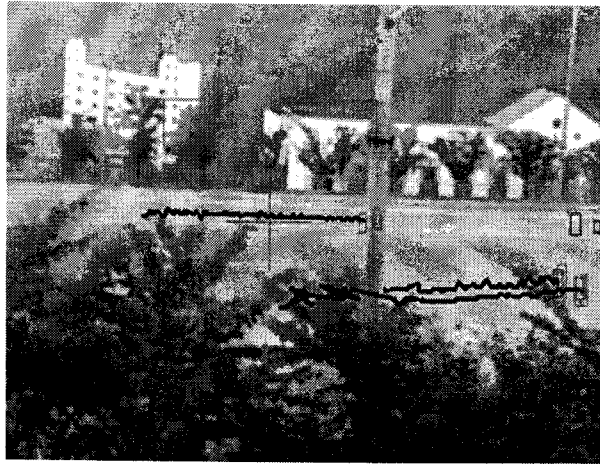
〈Fig. 6〉 Tracking object in outdoor environment

아래 그림은 실외환경에서 가려짐이 포함된 다수물체의 추적결과를 보여준다. 그림에서 궤적이 표시되는 움직임 객체는 지속적으로 정합 되고 있는 물체를 나타낸다.

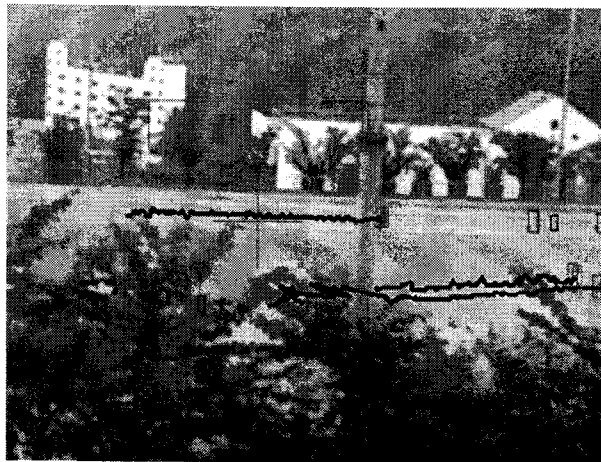
그림 a는 물체가 가려지기 직전, 그림 b는 물체가 가려진 상태, 그림 c는 물체가 다시 나타난 경우 성공적으로 물체를 추적하고 있음을 보여준다.



(a) 가려짐이 발생하기 직전



(b) 추적 중이던 물체가 가려진 경우



(c) 가려졌던 물체가 다시 나타난 경우

〈그림 7〉 가려짐이 발생하는 다수의 물체추적

〈Fig. 7〉 Tracking multiple objects including occlusion

실험결과 물체의 개수가 일정개수 이하이고 각각의 움직임이 독립적이면 성공적으로 정합을 지속해 나갈 수 있었지만, 물체의 개수가 많아지고 물체들 간의 충돌이 발생되면 이웃의 다른 객체로 정합되는 경우가 발생함을 알 수 있었다. 이는 추출된 객체의 크기가 너무 작고, 또 물체의 내부정보를 최소한으로 구성하여 정합에 사용할 충분한 정보를 제공치 못하는 것으로 판단된다.

V. 결론

본 연구에서는 프레임차이와 배경차이를 혼용한 방법으로 안정적인 배경모델을 형성하였고, 이 배경모델을 사용하여 움직임 물체를 탐지하고, 또 지속적으로 추적하기 위한 프레임워크를 제시하였다. 배경모델은 이동평균개념의 단일 가우시안 모델을 사용하여 매 프레임마다 배경모델을 갱신할 수 있도록 하여 외부환경변화에 적극적으로 대처할 수 있도록 하였고, 물체의 추적은 물체의 크기에 따라 물체 내부정보 및 위치정보의 중요도가 적응적으로 조절되게 하여 정합의 정확도를 높일 수 있었다. 아울러, 정합을 위해 현재 프레임에서의 객체의 위치를 칼만 필터를 이용하여 예측하도록 하여 가려짐이 있는 경우에도 효과적으로 물체를 추적할 수 있었다.

제시한 방법으로 한 개의 물체 뿐만 아니라 여러 개의 물체를 효과적으로 추적함을 입증하였으나, 아직도 물체들간의 충돌에는 효과적으로 대처하지 못하는 것으로 나타났다.

향후 연구에서는 가려짐이나 충돌에 보다 강건한 물체모델 및 이들의 정합방법, 즉 세밀한 유사도 측정 방법에 대한 연구가 필요하다. 아울러 이 연구를 확장하여 팬틸트 장비가 부착된 능동형카메라에 대한 추적방법에 대한 연구를 진행할 예정이다.

참고문헌

- [1] 지정규, 이광형, 김용균, 오해석, “객체 추적을 위한 적응적 배경영상 생성방법”, 「정보처리학회지」 B, 제10-B권, 제3호, 2003. 6, pp.329-338.
- [2] Collins, Lipton and Kanade, “A System for Video Surveillance and Monitoring,” *Proc. American Nuclear Society(ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, Pittsburgh, PA, April 25-29, 1999
- [3] 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*, vol.579, 1985, pp.72-78,
- [4] Teknomo, Takeyema and Inamura, “Frame-based Tracking of Multiple Objects,” *Proc. of the IEEE Workshop on Multi-Object Tracking(WOMOT'01)*, 2001.
- [5] Lepetit and Fua, “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey,” *Foundation and Trends in Computer Graphics and Vision*, vol.1, no.1, 2005, pp.1-89.
- [6] A.L. Gilbert, M.K. Giles, G.M. Flachs, etc., “A Real Time Video Tracking System,” *IEEE Trans. Pattern Analysis and Mach. Intel.*, vol.2, no.1, 1980, pp.47-56.
- [7] 오창윤, “물체와 배경의 분리를 이용한 물체 추적”, 포항공과대학교 대학원 석사학위논문,

1994. 2.

- [8] 육창근, 서창진, 양황규, 차의영, “모자이크 배경이미지 추출에 의한 자동 보행자 추적에 관한 연구”, 「한국정보과학회 춘계학술발표논문집」, 1998. 4.
- [9] 김계영, 이은주, 최형일, “차영상 분석에 의한 동작 정보의 추출”, 「한국정보과학회논문지」, 제21권, 제8호, 1994. 8.
- [10] 이희영, 최재영, 강동구, 김홍수, 차의영, 전태수, “배경영상을 이용한 목표물 추적에 관한 연구”, 「한국멀티미디어학회 1999년도 춘계 학술발표논문집(학술발표)」, 제2권 제1호, 1999, pp.386-390.
- [11] 임용호, 백중환, 황수찬, “퍼지 예측을 이용한 이동물체 추적”, 「한국항행학회논문지」, 제5권 제1호, 2001, pp.26-36.
- [12] 이상욱, “윤곽선 모델과 특징을 이용한 이동 물체 추적”, 「경상대학교 해양산업연구소보」, 제14권, 2001, pp.42-51.