실시간 칼라영상에서 객체추출 및 추적

최 내 원† · 오 해 석††

요 약

본 논문은 고정영역에서 움직이는 객체를 검출하기 위한 방법으로 배경영상과 입력영상의 차를 이용하여 객체를 추출하고 추출된 객체의 이동을 추적하는 방법에 대해 제안하였다. 객체를 추출하는 방법으로 고정영역에 새로운 객체의 위치를 파악하기 위해 전체 영상의 픽셀을 연산에 참여시키는 것이 아니라 영상의 테두리에 설정된 영역의 픽셀들만을 연산에 참여시켰다. 따라서 중앙영역이 연산에서 제외되어 객체추출의 시간을 효과적으로 단축시킬 수 있었다. 또한 설정영역에서 객체를 추출하기 위하여 시작위치를 먼저 파악하고 시작위치로부터 객체의 가로와 세로의 크기를 추출함으로써 객체의 영역을 검출하였다. 이동된 객체의 추적에는 추출된 중심좌표를 이용하였다.

Object Extraction and Tracking out of Color Image in Real-Time

Nae-Won Choi† · Hae-Seok Oh††

ABSTRACT

In this paper, we propose the tracking method of moving object which use extracted object by difference between background image and target image in fixed domain. As a extraction method of object, calculate not pixel of full image but predefined some edge pixel of image to get a position of new object. Since the center area is excluded from calculation, the extraction time is efficiently reduced. To extract object in the predefined area, get a starting point in advance and then extract size of width and height of object. Central coordinate is used to track moved object.

키워드 : 실시간 시스템(real-time system), 객체추출(object extraction), 객체추적(object tracking), 영상처리(image processing)

1. 서 론

최근 컴퓨터 비전이나 패턴인식과 같은 영상처리 관련 연구에서 객체인식은 그 응용분야가 다양하여 많이 연구되고 있으며, 객체를 인식하기 위해서는 객체의 추출이 선행되어야 한다. 특히 영상을 실시간으로 처리해야 하는 보안시스템에서는 객체를 인식하는 시간이 아주 짧아야 한다. 본 논문에서는 주위 환경의 변화가 없는 고정영역의 배경을 대상으로 하여 새로운 객체가 고정영역으로 입력되었을 때 객체의 위치를 찾아내어 객체의 영역을 추출하고 추적하는 방법을 연구한다.

차영상을 이용하여 객체의 위치를 찾아내고 이동을 추적하는 방법은 교통, 로봇제어, 의학, 군사 분야, 보안시스템 등 다양한 분야에서 이용되고 있다.

본 논문에서는 보안시스템의 구축에 도움이 될 수 있도록 실험실 환경에서 실시간으로 출입자의 탐지와 추적 시스템을 구현하였다.

본 논문은 주위 환경의 변화가 거의 없는 실험실이라는 특수한 환경으로 카메라를 통하여 입력되는 영상에 객체의 움직임이 없을때 배경 이미지로 결정하고, 이 배경영상과 입력영상의 차영상을 통하여 객체를 탐지하고 탐지된 객체를 이용하여 이동된 위치를 추적한다. 영상 전체를 대상으로 하는 객체 탐지 방법보다 일정영역에 대한 객체 탐지는 연산의 수를 줄임으로 처리속도를 빠르게 하고 이동된 물체의 정확한 위치와 모양을 추출해 낼 수 있다.

먼저 2장에서는 기존의 세 프레임을 이용하여 물체를 탐지하는 방법과 가변 탐색창을 이용한 물체의 추적에 대해 알아보고, 3장에서는 제안하는 객체의 추출방법에 대해 기술한다. 그리고 4장에서는 추출된 객체로부터 이동한 객체의 추적방법을 살펴보고, 5장에서 객체추출과 추적에 대한 실험을 통하여 제안방법을 평가한 다음, 마지막 6장에서 연구결론을 기술한다.

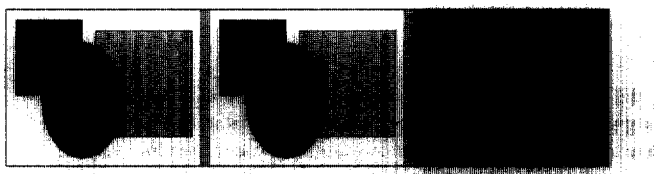
2. 객체추출과 추적 방법

배경영상과 입력영상을 이용하여 객체를 추출하는 방법에는 (1)배경이미지를 이용하는 방법과 (2)연속된 세 프레임의 차이를 이용하여 객체를 추출하는 방법이 있다[7].

(1)의 방법은 배경영상에서 입력영상의 차를 이용하여 객체를 추출하는 방법이다. 이 방법은 (그림 2-1)처럼 배경영

† 정 회 원 : 명지전문대 컴퓨터정보과 교수
†† 종신회원 : 숭실대학교 정보과학대학 교수
논문접수 : 2002년 7월 27일, 심사완료 : 2002년 12월 28일

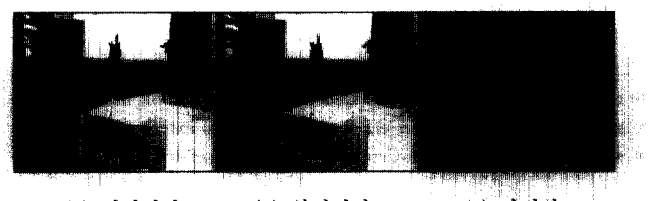
상이 시간의 차이에도 전혀 변함이 없는 영상, 즉 일반적인 이미지에 새로운 객체가 있을 때의 차영상을 구하였고, (그림 2-2)는 배경영상과 실시간으로 입력되어지는 영상과의 차영상을 구하였다.



(a) 배경영상 (b) 입력영상 (c) 차영상

(그림 2-1) 배경영상의 변화 없이 객체가 삽입된 경우

입력영상의 객체부분을 제외하고 모든 픽셀이 배경영상과 같다면 객체의 모양과 위치를 가장 빠른 시간에 찾아낼 수 있다. 하지만 실시간 영상은 배경의 거의 모든 픽셀 정보가 바뀌기 때문에 잡음이 많아 차영상 만으로는 객체를 추출하고 추적할 수 없다.



(a) 배경영상 (b) 입력영상 (c) 차영상

(그림 2-2) 실시간 영상에서의 차영상

차영상을 이용하여 객체를 추출할 경우 미디언 필터 등을 통하여 잡음을 제거해야 한다.

(2)의 방법은 다음과 같은 간단한 연산에 의해 오브젝트 이미지를 추출 할 수 있다[7].

$$A = F(t) - F(t-1), \quad B = F(t-1) - F(t)$$
$$Object(t) = A \cap B$$

이 방법은 간단히 객체를 추적할 수 있지만 객체가 아주 천천히 움직일 경우 두 영상간의 차이가 적어 차영상으로 구해진 이미지가 매우 작거나 차영상이 생기지 않을 수 있다[7]. 또한 (그림 2-2)에서처럼 차영상의 잡음에 영향을 많이 받으므로 정확한 객체가 추출되지 않을 수 있다.

객체의 추적을 위해서는 우선 추적 대상이 되는 객체의 형태를 배경으로부터 추출해 내고 객체의 위치를 파악하는 것이 필요하다. 에지 비교를 통한 객체의 위치를 찾는 방법은 추적 물체의 모델과 영상에 존재하는 추적 물체의 에지 간 거리를 계산해서 위치를 찾게 된다. 이전영상에서 객체를 찾게 되고 현재 영상에서 객체의 위치를 찾음으로 객체의 이동을 추적할 수 있다.

실시간 영상에서의 객체의 추적은 환경의 변화가 존재하는 연속영상으로부터 움직이는 객체를 추출 할 수 있어야 한다. 움직이는 객체의 추적을 위한 방법으로 3차원 모델기반의 방법[1], 영역기반의 방법, 윤곽선 기반의 방법, 특징기반의 방법 등이 있다. 3차원 모델 기반의 방법은 소수의 객체에 대해 정확성이 높은 모델과 궤적을 복원하는 것으로, 상세한 기하학적 물체의 모델이 주어져야 하는 단점이 있다. 영역기반 방법은 연속 영상에서 연결된 영역을 구하고 상관관계 측정을 이용하여 객체를 추적하는 방법으로 배경영상을 가지고 입력영상과의 차영상을 이용하여 객체를 검출한다. 윤곽선 기반의 방법은 객체의 윤곽선을 표현하고, 그것을 동적으로 갱신하면서 추적하는 방법이다[5].

3. 제안된 객체추출 방법

본 논문의 배경은 고정되어 있는 환경으로 배경영상과 입력영상의 차를 이용하여 객체의 위치를 파악하고 배경영상과 입력영상의 분할된 영역의 차가 허용치보다 높을 때 입력영상에 새로운 객체가 입력되었음을 알 수 있다.

(그림 3-1) 배경영상

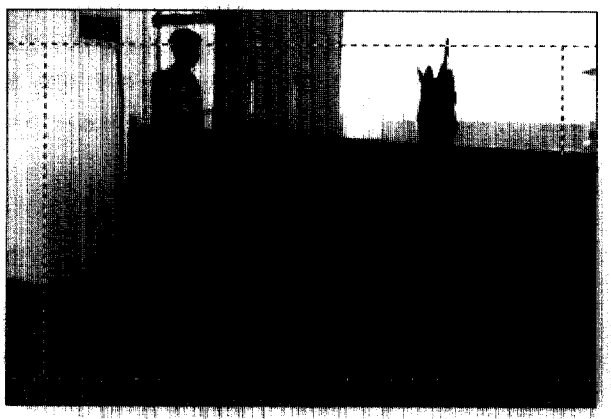
(그림 3-2) 입력영상

분할된 영역은 가로 세로에 위치한 가장자리의 연속된 픽셀 값으로 표현하며, 각 영역에서 R, G, B의 값을 검출하여 식 (1)을 만족하면 영상의 x, y의 위치에서 객체를 검출하기 위한 시작위치로 간주된다. 여기서 ORGB는 배경영상

이고 TRGB는 대상영상이다. 또한 α는 허용되는 영상의 차이다.

$$|ORGB(x,y) - TRGB(x,y)| \geq \alpha \quad (1)$$

배경영상의 좌표(x, y)에 대해 대상영상의 좌표(x, y)를 각각의 R, G, B에 대한 차를 구하고 허용되는 α 이상의 값을 가질 때 객체의 위치로 판별하여 객체검출의 시작위치 후보로 등록한다. 이때 차영상에서 발생하는 잡음을 배제하기 위하여 연속되는 픽셀들이 β만큼 있을 때 즉, 식 (1)을 만족하는 연속된 픽셀의 개수가 β보다 크거나 같을 때 후보로 등록되어 있는 (x, y)좌표가 객체의 시작위치가 된다. 시작좌표(ObjS(x))에서 객체의 가로 크기를 구하기 위해서 식 (1)을 만족하는 픽셀들을 구하여 마지막 픽셀의 위치 즉, 객체의 끝좌표(ObjE(x))를 얻는다.

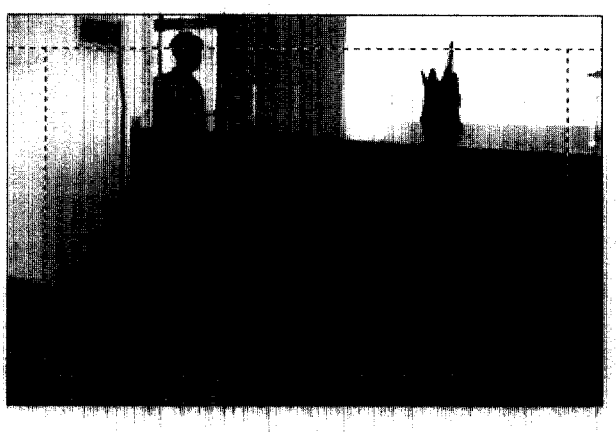


(그림 3-3) 입력영상의 분할영역[분할영역의 크기 = 20, β = 5]

객체의 가로시작위치와 끝 위치가 검출되면 세로의 크기를 검출하기 위하여 (그림 3-4)와 같이 객체의 가로길이(Obj(w))를 계산하고, 같은 방법으로 y 축에 대해 Obj(w)의 중앙에서 수직으로 객체의 높이(Obj(h))를 계산한다.

$$Obj(w) = ObjE(x) - ObjS(x) \quad (2)$$

$$Obj(h) = ObjE(y) - ObjS(y) \quad (3)$$

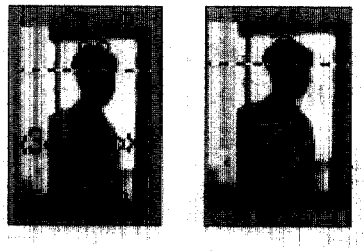


(그림 3-4) 객체의 길이와 높이 추출

객체의 중심좌표는 Obj(w)의 중간과 Obj(h)의 중간의 좌표이다. 이 중심좌표는 객체의 이동방향과 이동속도, 이동경로를 추적하기 위하여 식 (2)와 같이 계산된다.

$$ObjC(x, y) = [Obj(w)/2 + ObjS(x), Obj(h)/2 + ObjS)(y)] \quad (4)$$

(그림 3-5)와 같이 식 (5)와 식 (6)에서 계산된 값으로 식 (7)의 객체 크기를 얻는다.



(a) (b)

(그림 3-5) 객체의 크기 추출

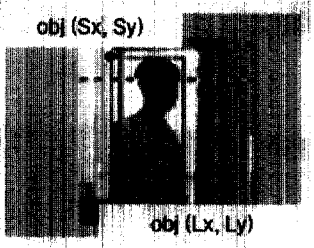
$$Obj(Lx) = Max(x_1, x_2, \cdots, x_n),$$
$$Obj(Sx) = Min(x_1, x_2, \cdots, x_n) \quad (5)$$
$$Obj(Ly) = Max(y_1, y_2, \cdots, y_k),$$
$$Obj(Sy) = Min(y_1, y_2, \cdots, y_k) \quad (6)$$

x축과 y축을 조밀하게 분할 할 때 객체의 정확한 크기를 추출할 수 있다. 이때 식 (5)의 n은 x축의 분할 개수이고 식 (6)의 k는 y축의 분할개수를 말한다.

따라서 객체의 크기는 다음 식과 같이 계산 될 수 있다.

$$Obj(Rect) = Rect(Obj(Sx), Obj(Sy), Obj(Lx), Obj(Ly)) \quad (7)$$

식 (7)의 Rect는 객체의 사각형 영역의 좌표를 말한다.



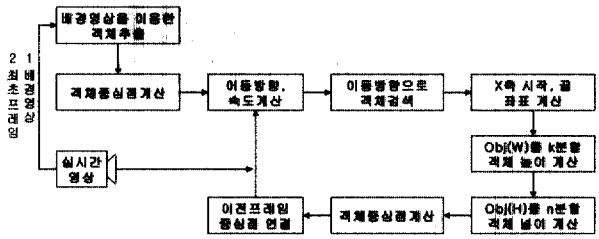
(그림 3-6) 객체영역

차영상을 이용한 객체의 추출에서 배경영상이 가지고 있는 각각의 R, G, B 값을 입력영상의 각 R, G, B 값과의 차이를 이용하여 객체의 최초 위치를 알아낸다. 최초의 위치에서 x축 방향으로 객체의 넓이를 계산한 후 계산된 객체의 넓이 중앙에서 y축 방향으로 확장하여 객체의 높이를 구한다. 객체의 높이를 n분할하여 객체의 최소 x좌표와 최대 x좌표를 구한다. 객체의 최소 y좌표와 최대 y좌표는 최소 x좌표와 최대 x좌표를 y축으로 k분할 확장하여 얻을 수 있다. x의 최소, 최대 좌표와 y의 최소, 최대 좌표를 객체의

크기로 나타내고 식 (4)에 의하여 객체의 중심좌표를 구한다. 이때 구하여진 중심좌표는 다음 프레임의 객체를 추출하기 위한 시작점으로 사용되며 다음프레임의 객체 중심좌표와 연결함으로 객체추적의 데이터로 사용된다.

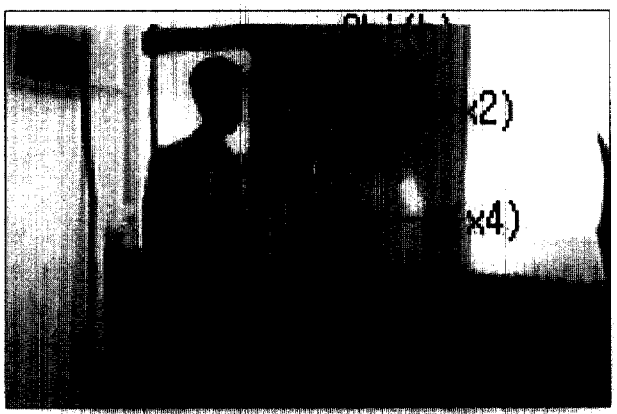
4. 객체추적

객체의 추적은 대상영상의 객체가 프레임의 변화에 따라 이동하는 위치를 추적하는 것이다. 객체의 추적을 위해서 3장에서 추출한 객체의 중심점을 이용한다. (그림 4-1)은 객체추적을 위한 블록도이다.



(그림 4-1) 객체추적 블록도

(a) 이동객체의 높이



(b) 이동객체의 넓이

(그림 4-2) 이동객체의 높이와 넓이

객체중심점에서 이전프레임으로부터 움직임 방향으로 1픽셀만큼 이동하여 새로운 좌표를 얻는다. 새로운 좌표와 배경영상과의 차를 구하여 식 (1)에 만족하는지 계산하고, 연속된 픽셀의 개수가 β보다 크거나 같을 때 최초의 x좌표는 이동객체의 시작좌표(ObjS(x))가 된다. 같은 방향으로 식 (1)이 만족하지 않을 때 까지 계속 진행하여 객체의 x끝좌표(ObjE(x))를 얻는다. 식 (2)에 의해 이동된 객체의 x축 중심을 구하고 y축 방향으로 확장하여 y시작좌표(ObjS(y))와 끝좌표(ObjE(y))를 구한다. 새로운 객체의 x좌표와 y좌표는 실제 객체의 크기를 계산하기 위해 (그림 4-2)(a)와 같이 x축을 기준으로 y에 대해 확장한다. w(x)를 k등분하여 각각의 높이를 식 (6)에 의하여 객체의 높이 Obj(h)로 결정한다. Obj(h)가 결정되면 x좌표의 넓이를 계산하기 위해 (그림 4-2)(b)와 같이 y축을 기준으로 x에 대해 확장하여 Obj(h)를 n 등분한 다음 각각의 길이를 식 (5)에 의해 객체의 길이 Obj(w)를 결정한다.

이때 k가 w(x)에 가까울수록 객체높이의 정확도가 높아지며 속도는 저하된다. 또한 n이 Obj(h)에 가까울수록 객체길이에 대한 정확도가 높아진다. Obj(w)를 y축에 대해 확장하고 Obj(h)를 x축에 대해 확장하여 만나는 좌표가 새로운 객체의 크기이다.

$$\text{New Obj(Rect)} = \text{Rect(Obj(Sx), Obj(Sy), Obj(Lx), Obj(Ly))} \quad (10)$$

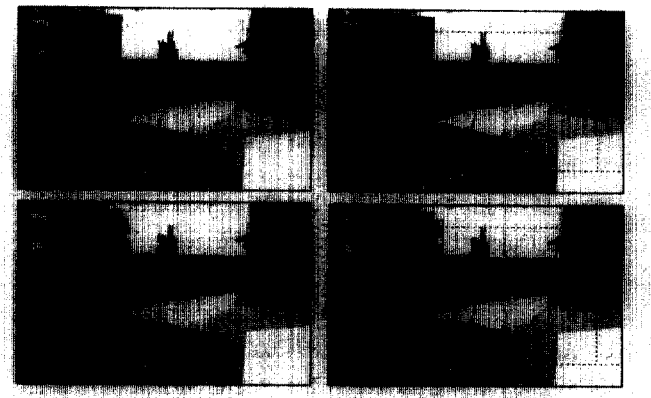
이때 이전프레임에서의 객체 중앙좌표와 현재 프레임에서의 객체 중앙좌표의 차를 구하여 객체의 이동방향과 속도를 구한다. 객체의 이동방향은 식 (11)과 같다.

$$\text{ObjD}(\alpha, \beta) = (\text{New ObjC(x)} - \text{Old ObjC(x)}, \text{New ObjC(y)} - \text{Old ObjC(y)}) \quad (11)$$

만약 α가 양수이면 객체는 왼쪽에서 오른쪽으로 움직였고, 음수라면 오른쪽에서 왼쪽으로 움직였음을 알 수 있으며, 0이면 움직임이 없음을 나타낸다. β가 양수이면 위쪽에서 아래쪽으로 움직였고, 음수라면 아래쪽에서 위쪽으로 움직였음을 나타내며, 0이면 움직임이 없음을 나타낸다. 객체 움직임의 속도는 $|\alpha|$와 $|\beta|$에 의해 움직임의 속도를 픽셀로 나타낼 수 있다. ObjD(4, 0)의 값을 갖는다면 객체는 왼쪽에서 오른쪽 4픽셀의 속도로 움직였으며 위쪽과 아래쪽으로의 움직임은 없음을 알 수 있다.

5. 실 험

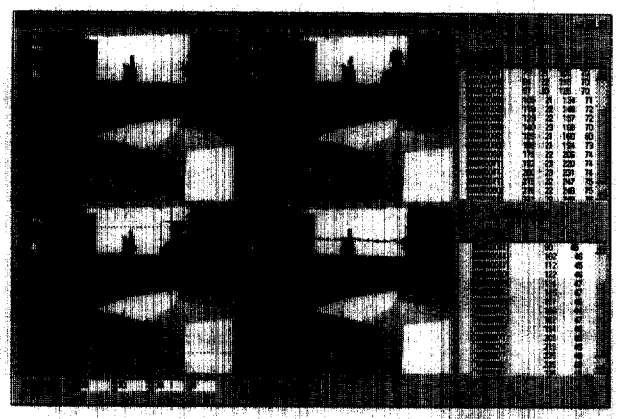
실시간 영상에서 객체의 추출과 추적에 대한 실험은 실험실에서 실제 실시간 영상으로 시뮬레이션해 보았다. (그림 5-1)과 같이 먼저 객체의 추출을 위해 배경영상을 획득한 후 최초로 움직임이 감지되는 프레임의 객체를 추출하였고, 객체추출의 정확성을 알아보기 위해 몇 프레임 후의 객체를 같은 방법으로 추출하였다.



(a) 입력영상의 객체 (b) 객체의 크기와 중심값 검출

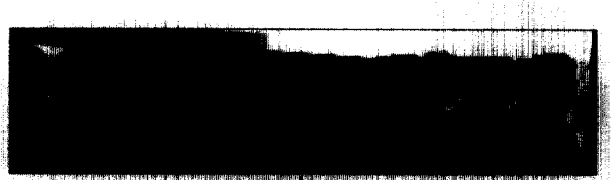
(그림 5-1) 입력영상에서의 객체 크기 및 중심값 검출

(그림 5-2)는 실시간 칼라영상의 객체를 추적하는 실험을 보이고 있다.

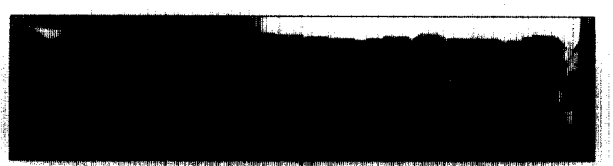


(그림 5-2) 실시간 영상에서의 객체 추적

먼저 객체를 검출한 후 객체의 중심값을 기준으로 다음 프레임의 객체를 찾아낸다. 찾아진 객체는 k, n 등분한 크기에 의해 가로와 세로의 최대 길이를 계산하게 되고 가로 세로 최대길이의 연결이 객체의 크기가 된다. (그림 5-3)은 실시간으로 움직이는 객체들의 움직임을 표현하였다. (그림 5-4)는 각 객체의 중심점을 점으로 표현하고 다음객체의 중심점과 연결함으로 객체의 추적을 보여주고 있다.

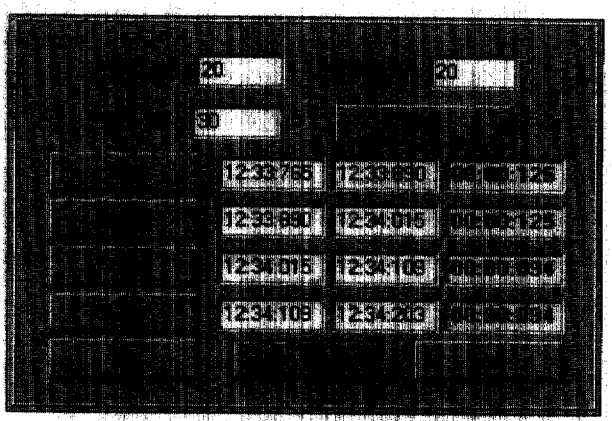


(그림 5-3) 객체의 이동



(그림 5-4) 이동객체의 추적

(그림 5-5)는 본 논문에서 제안한 테두리 영역에서의 객체검출을 위한 시간을 계산 한 것이다.



(그림 5-5) 테두리 영역의 객체검출 연산시간

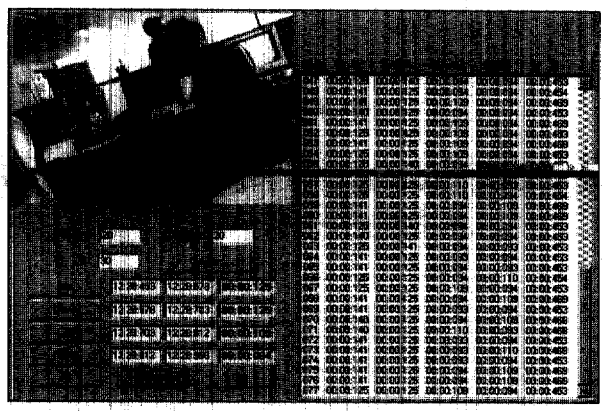
반복실험에 의한 데이터는 <표 5-1>에 기록되어 있다. 실험환경이 연구실 내부임으로 입력영상에서의 객체는 위쪽과 오른쪽에서 많이 추출되었다. 실제 시스템에서는 객체가 추출되었을 경우 객체의 크기 계산루틴을 수행하지만 타 시스템과 비교하기 위하여 객체가 추출되어도 설정된 테두리의 모든 영역을 검색하였다.

〈표 5-1〉 객체추출 시간

(단위 : ms)

횟 수	스캔 영역		비 교
	전체 영역	제안 시스템	
1	1.500	0.460	1.040
2	1.500	0.432	1.068
3	1.485	0.453	1.032
4	1.500	0.469	1.031
5	1.484	0.453	1.031
6	1.485	0.454	1.031
7	1.500	0.469	1.031
8	1.500	0.420	1.080
9	1.484	0.453	1.031
10	1.500	0.453	1.047

(320×240영상, 테두리 영역 높이 : 20 넓이 : 20, 객체추출 계산 포함)



(그림 5-6) 제안 시스템의 객체추출 시간계산

실시간 영상에서의 객체의 추출과 추적은 극히 짧은 시간에 객체를 찾아내고 움직임을 감지하여 추적해야 한다. 객체를 추출하는데 연산의 대상이 전체영상이라면 연산을 위한 많은 시간을 필요로 한다. <표 5-2>는 입력영상의 크기가 320×240일 때 객체추출 영역이 전체일 때와 제안한 시스템을 적용하였을 경우 연산양을 계산한 것이다.

〈표 5-2〉 객체추출시 계산에 참여되는 픽셀 수

계산에 참여하는 영역		계 산	계산량
전체영역에서 객체추출		320 × 240	76,800
제 안 시스템	첫 번째 영역에 객체가 존재	320 × 20	6,400
	두 번째 영역에 객체가 존재	320 × 20 + 240 × 20	11,200
	세 번째 영영에 객체가 존재	320 × (20 + 20) + 240 × 20	17,600
	네 번째 영역에 객체가 존재(최대)	320 × (20 + 20) + 240 × (20 + 20)	22,400

6. 결 론

본 논문에서는 실시간 영상에서 객체를 추출하는 방법과 추출된 객체의 추적방법을 제시하였다. 특정영역에서 객체를 추출하여 추출된 객체로부터 다음 프레임의 객체를 추적하는데 사용하였다. 객체의 추출은 배경영상과 입력영상 각각의 R, G, B의 차이에 의해서 수행되었고 객체의 넓이와 높이를 추출하였다. 객체의 추적은 추출된 객체의 중심좌표로부터 이동 방향과 속도를 계산하여 다음 프레임의 객체 위치를 알 수 있었고, 찾아진 객체의 중심좌표를 연결함으로 객체를 추적하였다.

제안한 객체의 추출 및 추적 알고리즘은 빠르게 수행되어야 하는 실시간 시스템에서 활용하기 위한 것으로 객체인식과 결합하여 객체의 움직임정보와 인식을 통한 보안시스템 등 응용 분야가 다양할 것으로 기대된다.

본 논문에서 언급하지 않은 다중객체의 추출 및 추적에 대해서는 객체의 이동방향과 속도, 특징정보의 조합으로 각 객체의 위치를 추적하기 위한 연구가 현재 진행 중에 있다.

참 고 문 헌

[1] D. Koller, J. Daniilidis and H. Nagel, "Model-based Object Tracking in Monocular Image Sequences of Road Traffic Sences," Int'l J. of Computer Vision, Vol.10, No.3, pp.257-281, Oct., 1993.

[2] J. Yang and A. Waibel, "A Real-Time Face Tracker," IEEE Workshop on Applications of Computer Vision, pp.142-147, 1996.

[3] S. M. Smith, "ASSET-2 : Real-Time Motion Segmentation and Object Tracking," Real-Time Imageing, Vol.4, pp.21-40, 1998.

[4] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley Inc., pp.189-200, 1995.

[5] 이상욱, "윤곽선 모델과 특징을 이용한 이동 물체 추적", J. Ins Marine Industry 14, pp.42-51, 2001.

[6] 서창진, 최은주, 양황규, 차의영, "동적 배경이미지 추출에 의한 자동 보행자 추적", 한국정보과학회 추계학술발표논문집, 제24권 제2호, pp.529-532, 1997.

[7] 이희영, 최재영, 강동구, 김홍수, 차의영, 전태수, "배경영상을 이용한 목표물 추적에 관한 연구", 한국멀티미디어학회 춘계학술발표논문집, pp.386-390, 1999.

[8] 박창준, 조상현, 최흥문, "잡음에 강건한 주목 연산자의 구현과 효과적인 다중 물체 검출", 전자공학회논문지, 제38권 제SP편 제1호, pp.89-95, 2001.

[9] 임용호, 백중환, 황수찬, "퍼지 예측을 이용한 이동물체 추적", 한국항행학회논문지, 제5권 제1호, pp.26-36, 2001.

최 내 원

e-mail : nwchoi@mail.mjc.ac.kr

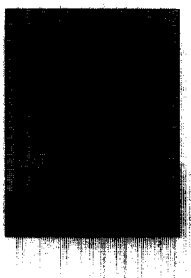
1980년 광운대학교 전자계산학과(공학사)

1986년 숭실대학교 전자계산학과(공학석사)

2002년 숭실대학교 컴퓨터학과(박사수료)

1983년~현재 명지전문대 컴퓨터정보과 교수

관심분야 : 멀티미디어, 영상처리, 데이터베이스



오 해 석

e-mail : oh@computing.ssu.ac.kr

1975년 서울대학교 응용수학과(이학사)

1981년 서울대학교 계산통계학과(이학석사)

1989년 서울대학교 계산통계학과(이학박사)

1982년~현재 숭실대학교 정보과학대학 교수

관심분야 : 멀티미디어, 데이터베이스, 영상처리