

선택적 개별 물체의 이동 추적 알고리즘 개발에 관한 연구

김 성 일[†]

요 약

본 논문에서는 움직임을 갖는 다중 물체 중에서 하나의 물체를 선택하고, 선택된 물체를 계속 추적하는 알고리즘을 제안하였다. 일반적으로 차영상을 이용하는 이동물체의 동작정보 추출방법은 주로 연속되는 영상내에서 일정한 영역의 영상특성을 정합하는 방법이 주로 사용되어왔다. 본 논문에서 제안한 동작정보의 추출방법은 연속영상간의 차연산을 한번 시행함으로써 얻어진 움직임영역을 기반으로 사용자에 의해 임의로 선택된 특정한 움직임영역을 추출하는 것이다. 특정한 모양이나 패턴을 인식하여 추적하는 것과는 달리 본 논문에서는 사용자에 의해 선택된 물체를 추적목표물체로 삼는 새로운 이동추적 방법을 제시하였다. 실험은 CCD 카메라로 입력된 실제 금붕어 영상을 이용하여 선택된 이동물체의 추적이 효과적으로 수행됨을 보였다.

A Study on the Development of Selectable and Individual Moving Object Tracking Algorithm

Sung-Il Kim[†]

ABSTRACT

The algorithm suggested in this article is to select one target object among multi moving objects and keep tracking that object continuously. General methods that match the specific area within continuous images is used as an extraction method to get the informations of moving object using differential image. The using method suggested in this article is that extracts the movement area obtained by performing difference operation once between continuous images. Unlikely, general methods that recognize and track specific form or pattern, the new movement tracking method suggested in this article takes the user-select object as the object to be tracked. In this experiment, the real goldfish image captured by CCD camera is used and it is shown that the selected goldfish is tracked efficiently.

키워드 : Tracking Algorithm, Moving Object, Selectable

1. 서 론

인간의 시각 구조는 움직임에 매우 민감하여, 움직임이 있는 곳에 초점을 맞추게 되며, 움직임이 있는 물체는 쉽게 감지할 수 있다. 이러한 인간의 시각적 움직임 정보처리 기능을 보다 효과적으로 실세계에 도입하기 위하여 하드웨어 및 소프트웨어 기술이 급속히 발전하고 있다.

연속적으로 입력되는 영상에서 구해지는 이동물체에 대한 특징을 이용하여 이동하는 물체의 움직임부분을 영역화할 수 있고, 물체의 인식, 움직임 분석등을 수행할 수 있으며, 순차적으로 처리되는 영상에서 움직이지 않는 배경부분을 동일한 데이터영역으로 처리하여, 동영상 전송분야의 데이터압축을 도입할 수 있다[1].

이동하는 물체의 움직임부분의 영역화란 영상에서 원하는 이동물체만을 영역화하여 처리할 데이터의 양을 줄이고, 원하는 부분을 추출하기 위한 것이다.

영역화가 수행된 후 이를 기반으로 대상물체의 움직임을 추적하는 방법으로 인접한 두 프레임의 차를 이용하는 영상차 방법(image subtraction)[11], 이전 프레임에서 특징(feature)을 뽑아 다음 프레임과 정합시키는 정합(matching) 방법[12], 각 화소에 대한 속도벡터를 나타내는 광류(optical flow)를 이용한 영역화 방법[13, 14] 등이 있다.

본 논문에서는 영상차방법으로 여러 개의 이동물체를 영역화하여 움직임을 추정하였다. 특히 본 논문에서 사용한 영역화의 특징은 차영상을 기반으로 움직임영역의 외곽선을 검출한 후 두 개의 좌표만을 이용하여 움직이는 물체를 영역화하였기 때문에 다른 방법에 비해 움직이는 물체의 고유한 특징추출이 어려운 단점이 있으나 간략한 알고리즘

※ 본 논문은 교내연구비로 지원되었음.

† 정 회 원 : 경원대학교 전기전자공학부 교수

논문접수 : 1999년 8월 17일, 심사완료 : 2001년 1월 3일

으로 빠른 시간내에 움직임영역을 추출해 낼 수 있다.

특히 어느 시점에서 어떤 물체를 추적할 것인가는 움직임 추정에서 반드시 필요한 요소로 볼 수 있다. 이런 관점에서 여러 개의 움직이는 물체들 중에서 원하는 물체를 추적하는 알고리즘을 제안한다.

2. 이동물체의 정보 추출 및 추적방법

움직임은 인간이나 동물이 배경 영상으로부터 관심 물체를 추출해 내는 유용한 단서가 된다[10].

입력영상 정보에서의 움직임 추출은 움직임이 있는 부분의 영역화로 표현할 수 있다. 입력영상 정보란 인간의 시각 체계와 유사한 카메라로부터 3차원 실세계 영상을 2차원의 평면으로 수집한 정보 데이터이며, 영역화란 서론에서 언급하였듯이 입력 영상 정보로부터 특징이 같은 부분들로 구분하는 과정이다.

주로 사용하는 방법은 움직임이 있는 영상 정보를 일정한 시간 간격으로 나누어 입력받은 후, 이전에 입력된 영상 정보와의 차이를 이용하는 방법이다. 그리고 영상 정보간의 차이를 어떻게 구별짓고 계산하느냐에 따라 여러 가지 방법이 제시되고 있다.

2.1 이동 물체 추적 알고리즘

지금까지 이동 물체의 움직임을 추정하는 알고리즘이 많이 개발되었는데 크게 이동 물체의 움직임 정보의 특성을 영역으로 얻는가, 구별되는 특징으로 얻는가에 따라 영역 기반 방법과 특징 기반 방법으로 분류할 수 있다.

2.1.1 영역 기반 방법

영역 기반 방법은 한 영상에서 특징을 포함한 일정 영역(area)의 영상 조각(image patch)을 가지고 다른 영상과 상관관계(correlation)를 이용하여 찾는 방법으로 다음과 같은 방법들이 있다.

① 영상차방법

우선 영역 기반의 영상차방법은 인접한 두 프레임의 차를 이용하여 영상 내에서 차이가 생기는 부분을 움직임 영역으로 간주하는 방법이다. 초기에 나온 움직임 추적 알고리즘으로서 구현하기 쉬운 반면 프레임과 프레임 사이에서 이동 물체가 겹쳤을 때 영역화된 이동 물체의 모양이 달라지거나, 카메라가 흔들렸을 경우 이웃하는 프레임들 사이에 대응하는 좌표에 대한 화소값이 판이하게 달라지므로 영역화의 성능이 많이 떨어진다[1]. 화소값에 따라 특정한 임계치를 기준으로 binary level로 또는 gray level로 양자화하는 방법이 있다.

② 블록 정합(Block Matching)법

영역 기반 블록 정합 방법은 한 영상에서 특징을 포함한

일정 영역의 부분 영상을 가지고 인접한 다른 영상과의 상관관계를 이용하여 찾는 방법이며, 그 상관관계는 화소 기준의 평균 제곱 차이값, 절대값 차이의 합, 또는 유사도 값을 가지고 비교하는 방법이며, 두 영상의 화소의 밝기값이 직접 정합에 사용되므로, 정합에 사용되는 화소들의 특징의 변화에 아주 민감하다. 이는 영상의 기하학적 왜곡이나 얼룩, 잡음등에 상당히 취약하며 특히 크기 변화나 회전이 존재하는 영상에서는 적용하기 어려운 점이 있다.

블록 정합의 대표적인 Full Search Block Matching Algorithm(FBMA)방식은 search할 블록에 대해 한 픽셀 단위로 블록 전체에 대한 움직임을 search하는 방식으로, 다른 블록 정합방법들에 비해서 한 픽셀 간격으로 움직임을 추정하기 때문에 가장 최적의 블록을 찾아낸다. 단점으로는 영상의 특성에 상관없이 움직임 추정에 동일한 탐색 영역을 설정함으로써 많은 계산적인 낭비를 초래한다[3].

그 외에 TSS(Three Step Search)알고리즘이나 CSA (Cross Search Algorithm)등은 계산량을 줄여 빠르게 블록 정합을 수행하게 하지만 예측 오차가 커진다는 단점이 있다.

2.1.2 특징 기반 방법

① 광류(Optical Flow)에 의한 방법

Optical flow 방법은 영상에서의 밝기 부분의 변화와 연관된 2-D velocity field이다. 실제로 3차원 공간상에서 물체의 움직임 추정을 할 경우에 motion field를 구하기가 상당히 힘들기 때문에 2-D image plane에서의 optical flow를 구하여 근사적으로 3-D motion field를 구한다[9].

일반적으로 카메라로 얻는 영상에서는 각 화소를 이루는 광도만을 측정할 수 있는데, 광류를 구하는데 광도를 어떻게 이용하느냐에 따라 미분법(gradient)을 이용한 방법, 상관관계(correlation)를 이용한 방법, 에너지를 이용한 방법 등으로 분류한다[1].

② 직선을 이용한 방법

직선을 이용한 방법은 점을 이용한 정합보다 여러 이점을 가지고 있는데 점보다는 하나의 영상에서 직선의 수가 적고, 선 자체가 뚜렷한 특징을 가지고 있으므로 잠재적인 정합의 경우의 수를 줄일 수 있다. 실제로 직선들은 거주 지역의 항공사진이나, 건물의 실내 등 인간이 조형한 환경의 영상 등에서 빈번하게 생긴다[2].

③ 구석점(Corner Point)을 이용한 방법

특징 기반 방법으로 구석점(corner point)을 이용한 방법이 있다. 이 방법은 특징 플로우(feature flow)기반 정보로서 이동체의 형태를 특징짓는 구석점을 사용한다. 이동 물체의 형태를 특징짓는 요소로서의 구석점을 추출하고, 특징점 주변의 영역 특성을 고려하여 시간에 따른 특징점의 이동 벡터로 구성된 특징 플로우를 구한 후 특징점의 위치, 특징 벡터의 이동각 및 이동량을 매개변수로 하여 이들을

특징 공간상에서 군집화 함으로써 이동 물체를 추적하는 방법이다[4]. 이밖에 윤곽선 추출을 이용한 방법등이 있다.

3. 움직임 추출 및 선택된 이동물체의 추적

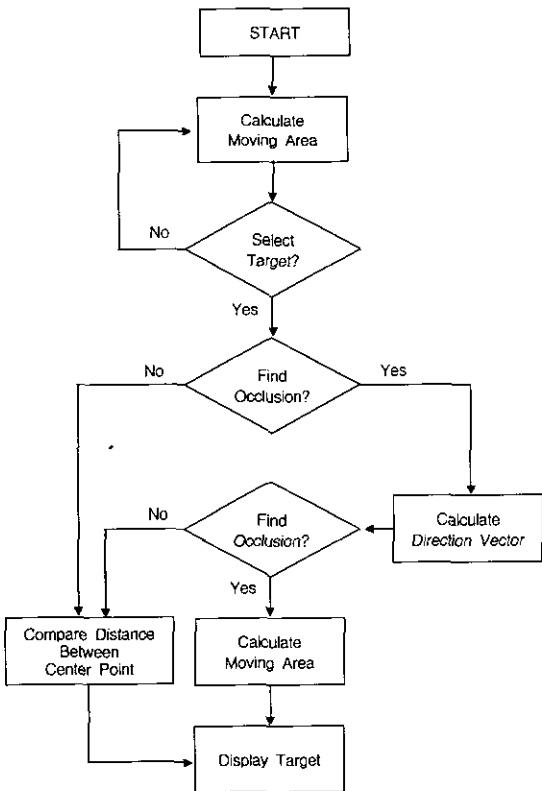
본 연구에서는 연속되는 영상내에서 영상차를 이용하여 복수의 이동물체에 대해 움직임영역을 추출해내어, 그중 임의의 움직임 영역에 대해 선택적으로 추적할 수 있는 알고리즘을 제안한다. 즉 영상내의 변화가 일어나는 순간에 포착된 물체를 추적한다면 각종 노이즈나 원치 않는 대상을 추적하게 될 확률이 높다고 할 수 있다.

그러므로 움직이는 대상물체를 추적하기에 앞서 대상물체를 선정할 수 있어야 한다.

기존의 움직이는 물체를 추출하는 알고리즘은 움직이는 물체들의 특징요소들을 찾는 방법이 주류로 이루었고, 또한 특징요소들을 찾는 과정에서의 계산시간을 줄이는 데에 초점이 맞추어져 있다.

그리고 최초로 움직임을 추적하기 위해 목표로 삼은 물체는 움직임이 검출된 영역에서의 임의의 물체이거나 유일한 물체였다.

그래서 제안하는 점은 최초에 여러 개의 임의의 움직이는 물체들중에서 사용자가 목표로 하는 하나의 물체를 선택해서 그 물체만을 추적하기 위해 다음과 같은 flow의 알고리즘을 제안한다.



(그림 1) 알고리즘 흐름도

최초에 입력된 두 영상의 차를 이용하여 움직임영역사각형(Moving Area Rectangle)을 구한 후 사용자의 입력 즉, 추적 목표물선택을 기다린다. 선택이 되어지면 중심점간 최소거리 및 방향벡터를 이용하여 선택된 목표를 추적을 수행하고 선택이 되어지지 않으며 계속해서 순차적으로 입력되는 영상내에 MAR을 표시하여 사용자의 입력을 기다리게 된다.

3.1 다중 이동물체의 움직임 정보 추출

입력된 영상에서 움직임 정보의 추출은 의미 있는 물체를 선정하기 위한 가장 기본적인 과정이다.

본 연구에서는 CCD카메라로 영상을 입력받아 Dipix 1000 프레임 그래픽을 사용하여 256 그레이레벨(gray-level)로 샘플링하여 만들어진 초당 2프레임의 연속된 캡처이미지이며 차영상기법으로 배경을 분리하고 움직임 정보를 추출한다.

먼저 최초에 입력된 두 프레임간에 차영상을 구한다.

$$I_d = |I_b(x) - I_n(x)| \quad (1)$$

여기서

- I_d : 두 영상의 차영상
- $I_b(x)$: 현재 프레임의 영상
- $I_n(x)$: 다음 프레임의 영상이다.

본 실험에 앞서 256×256크기의 오대산영상위에 다른 물체의 이미지를 합성하여 실험하였다. 합성된 이미지는 좌측 상단과 우측상단의 비행기모양이다.

(그림 2)에서 움직임이 있는 부분은 연속된 두 개의 입력 영상의 차이다. 이 움직임 영역을 데이터로 추출해 내기 위해 다음의 알고리즘을 수행한다.



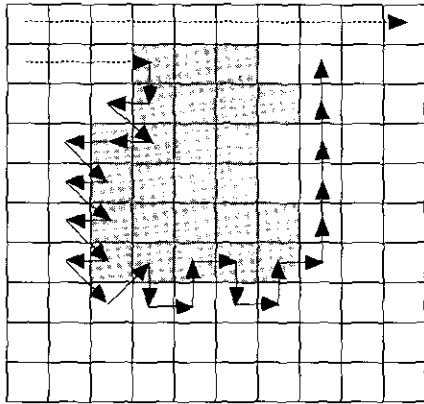
(그림 2) 두개의 입력영상에서의 차영상

먼저 (그림 3)에 나타낸 바와 같이 영상에서 각 픽셀값을 x축의 +방향, y축의 +방향으로 스캔하여 "0"이 아닌 값을 갖는 픽셀의 위치를 찾는다. 찾아진 픽셀의 위치를 스타트 포인트로 설정하고 그 위치부터 움직임영역을 찾는다.

시작점을 기준으로 각 픽셀의 값과 이전의 진행방향에 따라 6개의 방향으로 움직임의 영역을 찾는다.

움직임 영역은 위의 알고리즘으로 구해진 두 개의 좌표 값에 의한 사각형이다. (그림 4a)에서처럼 두 영상의 차이에서 나타나는 부분이 MAR로 설정된다.

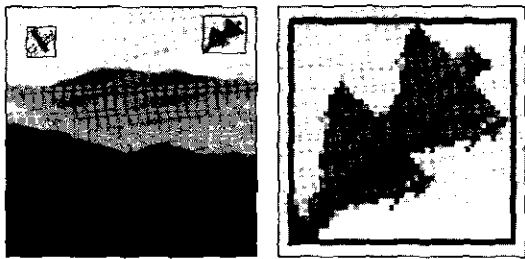
움직임 영역이 추출되었으면 그 움직임 영역 중에서 추적하기 위한 임의의 영역을 사용자가 마우스입력으로 선택하였다.



(그림 3) 시작점 설정

<표 1> 움직임영역 추출 스캔알고리즘표

	다음 진행방향		이전방향과 현재상태
1	↓	down	· 최초로 시작할 때 · "←"이고 "0"이 아닐때 · "↘"이고 "0"이 아닐때
2	↑	up	· "↘"이고 "0"일때 · "↑"이고 "0"일때 · "→"이고 "0"일때
3	→	right	· "↓"이고 "0"일때 · "↑"이고 "0"이 아닐때
4	←	left	· "↓"이고 "0"이 아닐때 · "←"이고 "0"이 아닐때 · "↘"이고 "0"이 아닐때
5	↗	right_up	· "↘"이고 "0"일때
6	↙	right_down	· "←"이고 "0"일때



(a) (b)
(그림 4) 움직임영역의 사각형(MAR)

3.2 선택된 이동물체의 추적

위에서 선택된 이동물체를 추적하기 위해서는 특징추출이 필요하다. 본 연구에서는 MAR의 공통부분추출과 무게 중심간의 최소거리, 그리고 방향벡터를 이용하였다.

MAR의 공통부분추출은 $Rect_i$ 를 현재프레임의 MAR이라고 하고, $Rect_{i+1}$ 를 다음프레임의 MAR이라하면

$$Rect = Rect_i \cap Rect_{i+1} \quad (2)$$

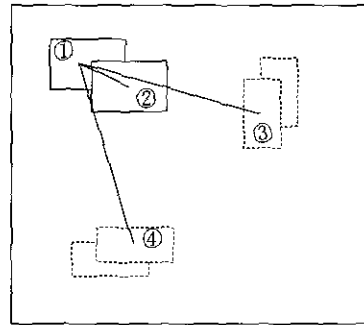
이 되고, MAR의 좌표를 (X_1, Y_1) 와 (X_2, Y_2) , 중심점을 (X_c, Y_c) 이라 하면

$$(X_c, Y_c) = \left(\frac{X_1 + X_2}{2}, \frac{Y_1 + Y_2}{2} \right) \quad (3)$$

$$D_c = \sqrt{(X_c - X_{c1})^2 + (Y_c - Y_{c1})^2} \quad (4)$$

에서 최소거리의 D_c 를 갖는 움직임영역을 선택하였다.

(그림 5)는 연속된 이동물체의 입력영상을 MAR로 표시한 그림이다. 여기서 직선으로 표시된 ①번 MAR이 사용자에게 선택된 영역이라고 가정하면 수식(2)에서처럼 공통부분을 조사한 후 ②, ③, ④번 MAR간에 D_c 를 비교하여 최소인 D_c 를 갖는 ②번 MAR을 선택하게 된다.



(그림 5) 공통부분과 중심점간 최소거리를 이용한 이동물체의 추적

3.3 겹쳐짐 문제(Occlude Problem)

움직임영역의 추출 및 추적은 물체의 겹쳐짐의 상황을 고려하지 않을 수 없다. 그래서 이러한 경우에는 MAR의 관계와 중심점간의 최소거리만으로는 하나의 움직임영역을 추적하기가 어려우며 잘못된 판단으로 다른 물체를 추적대상으로 인식하기가 쉽다. 그래서 또 다른 알고리즘을 적용하여야 한다.

본 연구에서 제안하는 알고리즘은 이동하는 물체가 서로 겹쳐졌음을 인식하고, 그 이후의 시점에서 중심점간의 방향 벡터를 적용한다.

먼저 이동물체가 서로간에 겹쳐짐을 인식하는 경우는

$$Logic(Rect1_{i-1} \cap Rect_i) = R_1$$

또는 $Logic(Rect_i \cap Rect1_{i+1}) = R_1,$

$$Logic(Rect2_{i-1} \cap Rect_i) = R_2$$

또는 $Logic(Rect_i \cap Rect2_{i+1}) = R_2,$

$$Logic(Rect3_{i-1} \cap Rect_i) = R_3$$

또는 $Logic(Rect_i \cap Rect3_{i+1}) = R_3 \dots\dots$

이라 할때

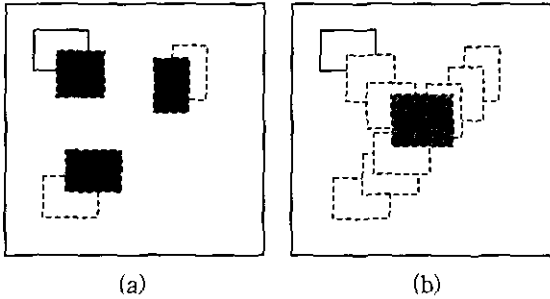
$$\sum_{n=1}^m R_1 + R_2 + R_3 + \dots + R_n \geq 2 \quad (5)$$

인 경우이다.

(그림 6a)에서 3개의 이동물체에 대한 연속영상을 나타내었고 b에서는 이동물체들이 진행하여 중간 지점에서 겹쳐졌음을 나타낸다. 이때에 수식(5)에서와 같이

$$\sum_{n=1}^3 R_1 + R_2 + R_3 = 3 \quad (6)$$

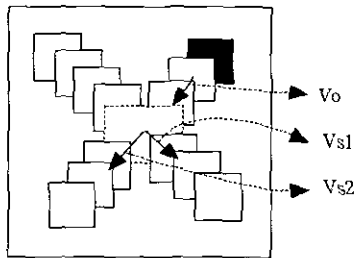
이므로 이동물체가 서로 겹쳐졌음을 인식한다.



(그림 6) 움직임영역의 겹쳐짐 인식

그리고 난 후 우리의 관심은 이 겹쳐진 물체들이 분리된 후에 어떤 것을 선택할 것인가이다.

(그림 7)에서 어두운 영역이 사용자에게 의해 선택되어진 부분이다. 계속하여 프레임을 입력받으면서 사용자에게 의해 선택된 영역과 일치하는 부분을 찾는다.



(그림 7) 각 프레임에 따른 움직임영역의 벡터표현

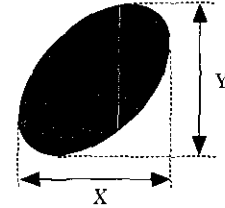
점선의 움직임영역이 계산되어진 후 2개로 겹쳐지게 되는 순간에 방향벡터 V_0 를 계산한다. 그리고 다시 움직임영역이 겹쳐지는 순간에 방향벡터 V_{s1} 과 V_{s2} 를 계산하고 V_0 와 비교한다. 비교치가 적은 벡터를 선택하고 움직임을 계속 추적한다.

이러한 근거로써 물체의 운동은 짧은 시간에 급격한 운동벡터의 변화가 없다는 가정을 둔다.

4. 실험 결과 및 분석

움직임영역의 추적을 위하여 본 논문에서는 합성영상과 실제영상을 사용하였다. 합성영상은 256×256크기의 오대산

을 배경으로 하고 임의의 두 비행기영상을 오대산영상 위에서 움직이게 하였는데 하나는 X, Y축을 따라 '+'값을 갖는 방향으로, 즉 좌상에서 우하로 이동하였고, 다른 하나는 X축은 '-'값을 Y축은 '+'값을 갖는 방향으로 이동하게 하여 중간 지점에서 두 물체가 겹쳐지게 하였다.



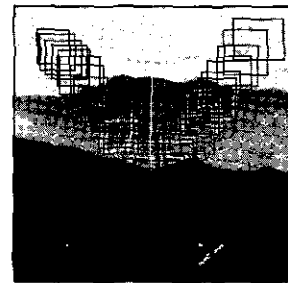
(그림 8) 이동 물체의 크기

두 물체의 이동속도는 임의의 가속도를 갖게 하고 (그림 8)에서 처럼 물체의 폭에 따라서

$$V = \sqrt{X^2 + Y^2} \quad (7)$$

의 속도 이하로 제한하였다.

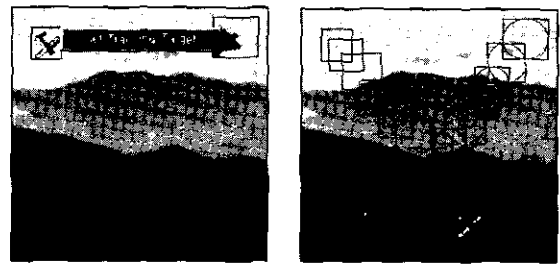
실제영상은 Indoor상태에서 어항속에 세 마리의 금붕어 영상을 이용하였다. 이 영상은 CCD카메라를 이용하여 Dipix사의 XPG-1000 프레임 그레머를 사용하여 8비트 그레이 레벨(gray-level)로 샘플링하여 만들어진 640×480의 크기를 갖는 초당 2프레임의 연속영상으로 구성되어 있다.



(그림 9) 합성영상의 이동물체 흐름

합성영상과 실제영상을 이용하여 각 프레임별로 이동 물체가 중복되어지게 나타남을 (그림 9)에 나타내었다.

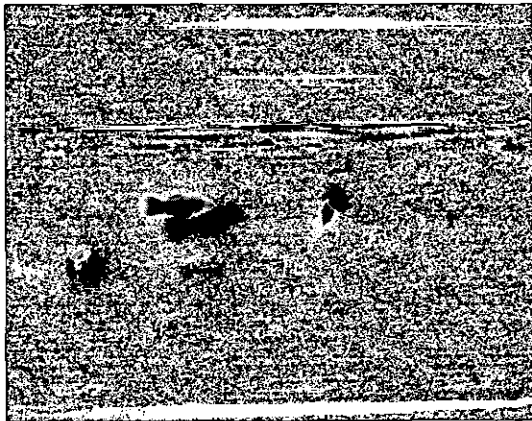
(그림 9)에서 나타난 움직임영역 중 사용자가 선택한 하나의 움직임영역을 계속해서 추적하게 된다.



(a) (b)
(그림 10) 합성영상의 물체선택과 추적

(그림 10a)에서 움직이는 두 개의 물체 중 왼쪽-위의 물체를 선택하여 (그림 10b)에 추적결과를 나타내었다.

다음은 실제 영상을 이용한 실험 결과이다. 실제 영상은 어항 속에 금붕어 3마리가 움직이는 모습을 이용하였다. (그림 11)은 인접한 두 영상의 차영상을 반전시킨 그림이다. 노이즈로 인해서 움직임 영역 이외에 픽셀값이 '0'이 아닌 부분이 나타나 있음을 알 수 있다.



(그림 11) 차영상의 반전

(그림 11)에서 가운데 부분에 두 마리의 금붕어 모습이 잡혀있고 왼쪽으로 나머지 한 마리가 위치해 있다. 그 위쪽으로 어항의 수면 부분이 나타나 있다. 합성 영상에서는 나타나지 않지만 실제 영상에서는 입력장치인 카메라의 성능과 자연 환경 조건에 따라서 영상에 많은 노이즈가 생길 수 있다.

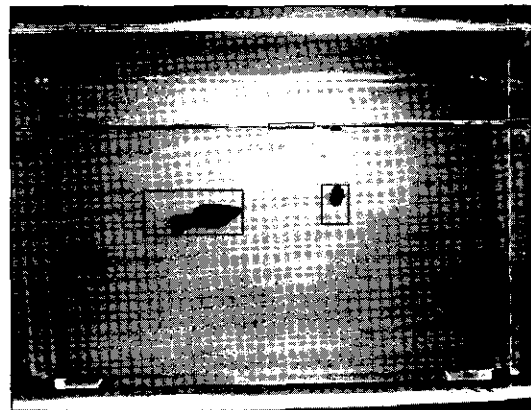
(그림 11)영상을 가지고 움직임 영역 사각형을 구할 수는 없으므로 low-pass 필터링 과정을 거친다. (그림 12)는 위의 그림을 low-pass 필터링 한 영상이다.



(그림 12) Low-Pass 필터링을 한 영상

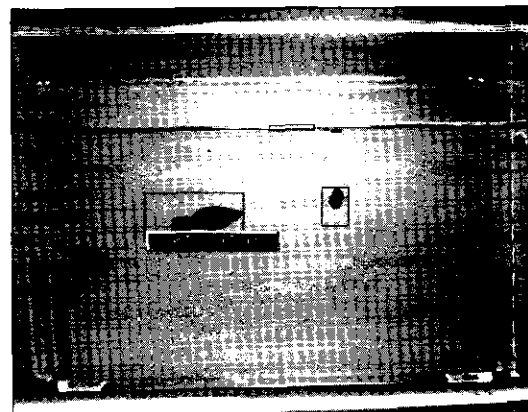
위의 그림에서 움직임 영역 사각형을 찾는 알고리즘이 수행된다. 제일 먼저 찾아지는 시작점은 예상하지 못했던 영상의 위-왼쪽 부분에 나타난 수면의 영역에서 시작하게

된다. 어항의 물결이 흔들렸거나 각종 부유물이 카메라 영상에 잡힌 것으로 추측된다. 이렇게 해서 구해진 첫 번째 프레임과 두 번째 프레임 사이에서의 움직임 영역 사각형이 (그림 13)이다. 사용자가 움직임 영역을 선택할 수 있도록 두 번째 프레임의 영상 위에 움직임 영역 사각형을 그렸다.



(그림 13) 실제 영상의 움직임 영역 사각형

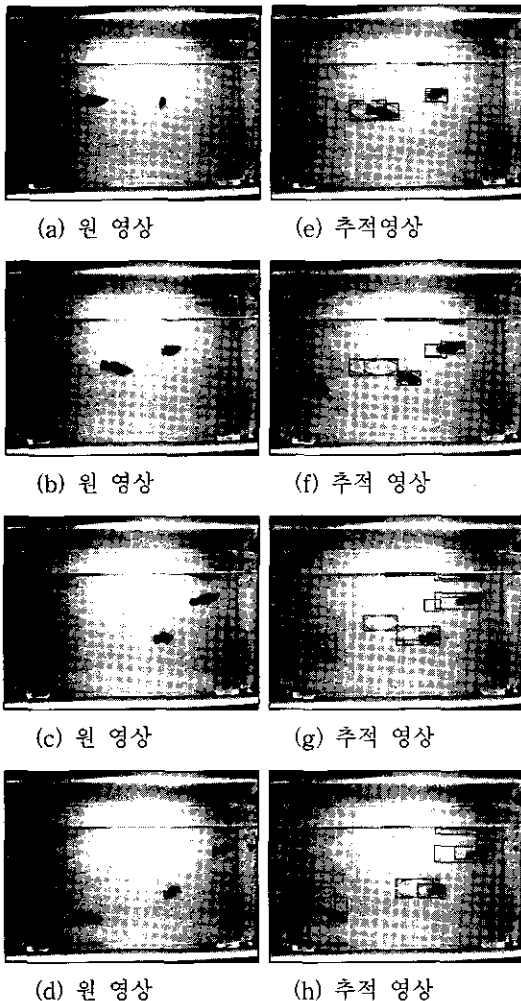
(그림 14)에 여러 개의 움직임 영역 사각형 중에서 사용자가 선택한 움직임 영역 사각형을 나타내었다. 이때 나타나는 표시는 마우스 오른쪽 버튼을 클릭 했을 때 나타나는 메시지이다.



(그림 14) 실제영상의 이동물체 선택

(그림 15)에는 (그림 10a)에서처럼 선택되어진 움직임 영역 사각형안에 타원을 표시하여 현재 선택되어진 물체가 순차적으로 추적되어짐을 보였다. 가장 오른쪽의 금붕어 움직임을 살펴보면 위로 점차 올라갈 때 수면에서 반사되는 이미지와 어항의 오른쪽 끝면에서 반사되는 이미지가 함께 잡혀있다. (그림 16)은 선택된 물체가 추적되어진 상황을 자취로 표시하였고 여기서 나타나듯이 추적 대상으로 선택되어진 금붕어 이외에 나머지 두 마리에서 움직임 영

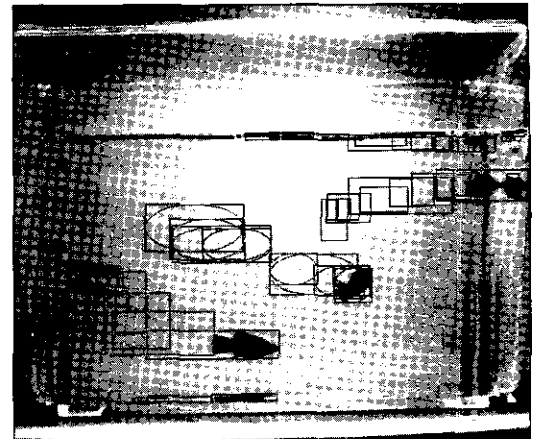
역 사각형이 한 물체에서 두 개 이상으로 분리되어 나타남을 알 수 있다. 이것은 필터링 과정에서 마스크와 기준값에 따라서 변화될 수 있는데, 노이즈가 많으면 하나의 물체가 두 개 이상으로 분리되어질 수 있다. 그러나 다행히 미세한 움직임을 갖는 상태에서 눈에 띄는 움직임을 갖는 상태로 전환된다면 분리되어지는 문제는 대상물 추적 성능에 크게 영향을 미치지 않는다.



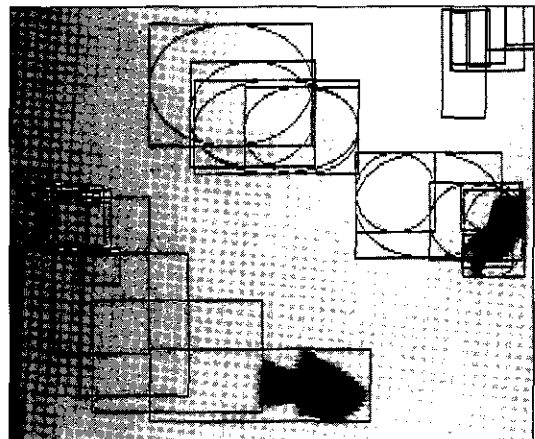
(그림 15) 실제영상의 선택된 물체추적(부분영상)

의 금붕어 움직임을 살펴보면 위로 점차 올라갈 때 수면에서 반사되는 이미지와 어항의 오른쪽 끝면에서 반사되는 이미지가 함께 잡혀있다. (그림 16)은 선택된 물체가 추적되어진 상황을 자취로 표시하였고 여기서 나타나듯이 추적 대상으로 선택되어진 금붕어 이외에 나머지 두 마리에서 움직임 영역 사각형이 한 물체에서 두 개 이상으로 분리되어 나타남을 알 수 있다. 이것은 필터링 과정에서 마스크와 기준값에 따라서 변화될 수 있는데, 노이즈가 많으면 하나의 물체가 두 개 이상으로 분리되어질 수 있다. 그러나 다행히 미세한 움직임을 갖는 상태에서 눈에 띄는 움직임을

갖는 상태로 전환된다면 분리되어지는 문제는 대상물 추적 성능에 크게 영향을 미치지 않는다.



(그림 16) 실제영상의 선택된 물체추적(자취)



(그림 17) 추적물체의 표시

<표 2> 기존알고리즘과 제안알고리즘의 특성차이

	기존 방법	제안한 방법
알 고 리 즘 특 성	• 특징을 포함한 부분영역의 화소값에 대한 상관 관계를 추적하거나 각 화소값이 갖는 특징값을 추출해야함	• 움직임영역 사각형의 공통부분추출 및 무게중심점간의 최소거리와 방향벡터 계산
	• 추적물체 선택의 기준이 없음	• 사용자에게 의한 임의의 추적물체선택 및 변경이 가능
	• 일정영역에 대한 모든 화소값 계산	• 일정영역에 대한 외곽의 화소값만을 계산
	• 프레임간 일정영역내의 화소값 비교계산 및 화소의 특성 비교에 따른 계산량 증대	• 프레임간 차영상계산으로 움직임영역을 검출하여 계산량 감소
	• 일정한 영역에 대한 화소값들의 비교 및 평균 연산, 미분연산들로 인한 계산량의 증대	• 임의 영역의 외곽선검출 및 영역겹침 계산에 따른 계산량 축소
• 하나의 영상에 대해서 2회이상의 전체화소 스캔이 필요	• 하나의 영상에 대해서 1회만의 전체화소 스캔 필요	

〈표 3〉 기본연산식과 계산량의 차이점

기 본 연 산 식	
기 존 방 법	<ul style="list-style-type: none"> • 최초 전영역 탐색 : n*m픽셀을 하나의 블록으로 설정시 N*M픽셀의 이미지를 목표물체의 블록수(k)만큼 정합 -> (n*m픽셀 block)*k*(N*M이미지) • 탐색영역 설정 case1 x+방향 = ((n*m픽셀 block) + a)*2 case2 x-방향 = ((n*m픽셀 block) + a)*2 case3 y+방향 = ((n*m픽셀 block) + a)*2 case4 y-방향 = ((n*m픽셀 block) + a)*2 • 움직임영역(움직임추정) 설정 (n*m픽셀 block)*k*(case1+case2+case3+case4)
제 안 한 방 법	GetDifferenceImage = I1(x, y) - I2(x, y) GetDifferenceImage = N X M pixels of First Image - N X M pixels of Second Image GetMovingArea = GetDifferenceImage (N X M) X Tracking Edge Points (I X J)
계 산 량	
기 존 방 법	$n*n(\text{최초전영역탐색}) \leq X \leq n*n+2*n*n$ (n = 영상을 구성하는 픽셀의 개수)
제 안 한 방 법	$10 n \leq x \leq n^2 + 10n$ (n = 영상을 구성하는 pixel 의 개수)

5. 결 론

본 논문에서는 이동물체 추적을 위한 새로운 표적 영상 영역화 방법으로 차영상을 이용한 다중이동물체환경에서 선택적 이동물체의 추적방법을 제안하였다.

본 논문이 목적하는 바는 기존의 시스템이 갖는 계산량 및 시스템의 복잡도를 개선한다는 취지보다는 이동물체추적의 새로운 접근방법을 제시하는데 중점을 두었다. 따라서 본 논문의 방법으로 영상을 처리할 경우 기존의 시스템과는 달리 추적물체 설정을 특정 데이터에 의한 목표물이 아니라 사용자 스스로에게 맡겨서 그 물체를 추적하도록 하였다.

합성 영상 실험에서는 움직임 영역 추출, 움직임 영역의 선택 및 추적 모두 만족할 만한 결과가 나왔다. 그러나 실제 영상 실험에서는 갑자기 빠른 움직임을 보이는 경우 앞에서 설정했던 움직임 영역이 겹쳐져야 한다는 가정에 위배되어 실험 자료로 취하기가 어려웠다.

총5회에 걸쳐 50개의 연속된 금붕어 영상을 캡처한 후 움직임 영역 추출과 움직임 영역의 선택 및 추적실험결과 83%의 정확도를 보였다.

개선되어야 할 부분은 사람이 움직임을 연속적으로 느끼게 되는 초당 24프레임 정도의 디지털 영상 입력장치가 필수적이며 또한 내부적으로 계산 속도가 그에 맞게 빨라져야하겠다. 그리고 물체의 형상이 반사되는 경우와 다른 종류의 물체가 함께 움직이는 환경에서 추적이 가능하도록 패턴인식이 접목되어야 할 것이다.

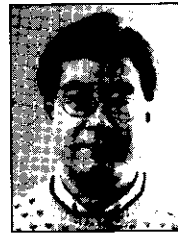
또한, 좀 정확하고 예측치 못한 상황에 대처하기 위하여,

그리고 영상 입력의 비교적 상태에서도 유연한 추적성능을 보이기 위해서는 블록 정합 및 광류를 부분적으로 이용하는 방법이 연구되어야 할 것이다.

참 고 문 헌

- [1] 김계영, 이은주, 최형일, “차영상 분석에 의한 동작 정보의 추출”, 정보과학회논문지, 제21권 제8호, pp.1480-1489, 1994.
- [2] 최중용, 이경행, 김태균, “위치결정을 위한 직선특징기반 스테레오 영상의 정합”, 정보과학회논문지, 제22권 제10호, pp. 1445-1458, 1995.
- [3] 이재승, 최우영, “가변탐색영역을 이용한 움직임추정 알고리즘”, 대한전자공학회, 추계종합학술대회논문집, 제20권 제2호, pp.779-782. 1997.
- [4] 이규원, 정우성, 이원열, 송홍엽, 박규태, “비매개변수 모델 기반 특징 플로우를 이용한 이동 물체의 검출 및 추적”, 대한전자공학회, 추계종합학술대회논문집, 제20권 제2호, pp.775-778. 1997.
- [5] 김명진, 한준희, “형태변화의 연속성을 이용한 이동 물체의 추적”, 정보과학회논문지, 제20권 제5호, pp.686-698, 1993.
- [6] 변재용, “Optical flow의 sequence coding의 적용에 관한 연구”, 석사학위논문, 광운대, 1996.
- [7] 김성일, “가중치 행렬에서 Euclidean 거리 측정법을 이용한 문자인식에 관한 연구”, 제1회 문자인식 워크샵, pp.169-172, 1993.
- [8] 송경근외 6인, “윤곽선 추출방법을 이용한 Core Skeleton검색 알고리즘”, 한국정보과학회 봄 학술발표회, Vol.20, No.1, pp. 175-178, 1993.
- [9] B. K. P. Horn and B. G. Schunck, “Determining Optical Flow,” Artificial Intelligence, Vol.17, pp.185-204, 1981.
- [10] H. -H. Nagel and W. Enkelman, “An investigation of smoothness constraints for the estimation of displacement vector fields image sequences,” IEEE Trans. PAMI Vol.8, No.5, pp.565-593, 1986.
- [11] S. A. Sugimoto, H.Matski, and Y.Ichioka, “Implementation of Tracking and Extraction of Moving Objects in Successive Frames,” Appl. Optics Vol.25, No.6, pp.990-996, 1986.
- [12] M. Sonka, V. Hlavac and R. Boyle, “Image Processing, Analysis and Machine Vision,” CHAPMAN & HALL COMPUTING, 1993.
- [13] Ralph Hartley, “Segmentation of optical flow fields by pyramid linking,” Patt. Recog. Lett. Vol.3, No.4, pp.253-262, 1985.
- [14] D. W. Murray and B. F. Buxton, “Scene segmentation from visual motion using global optimization,” IEEE Trans PAMI Vol.9, No.2, pp.220-228, 1987.
- [15] A. Singh, “Optical Flow Computation,” IEEE Computer Society Press, 1991.

- [16] Gregory A. Baxes, "Digital Image Processing," John Wiley & Sons, 1994.
- [17] S. Yamamoto, Y. Mae, Y. Shirai, and J. Miura, "Realtime multiple object tracking based on optical flows," Proc. Robotics and Automation, Vol.3, pp.2328-2333, 1995.
- [18] M. V. Srinivasan, "Generalized gradient schemes for the measurement of two-dimensional image motion," Biological Cybernetics, Vol.63, pp.421-431, 1990.
- [19] T. Uhlin, P. Nordlund, A. Maki, and J. O. Eklundh, "Towards an active visual observer," 5th International Conference on Computer Vision, pp.679-686, 1995.



김성일

e-mail : ksi@mail.kyungwon.ac.kr

1977년 연세대학교 전자공학과 졸업(학사)

1977년 한국과학기술연구소(KIST) 연구원

1980년 연세대학교 대학원 전자공학과

(공학석사)

1984년 연세대학교 대학원 전자공학과

(공학박사)

1992년 미국 IBM Almaden 연구소 초청과학자

1987년~현재 경원대학교 전기전자공학부 교수

관심분야 : Mobile IP, TCP/IP, 디지털신호처리, 패턴인식