

논문 2005-42SP-3-3

# 움직임 추정 능동 방사선 기반 고속 객체 추적 (Motion-Estimated Active Rays-Based Fast Moving Object Tracking)

라 정 중\*, 서 경 석\*\*, 최 흥 문\*\*\*

(Jeong-Jung Ra, Kyung-Seok Seo, and Hung-Moon Choi)

## 요 약

객체 중심점에서 움직임을 추정하여 빠르게 이동하는 객체의 윤곽선을 추적 할 수 있는 알고리즘을 제안하였다. 방사상 표현(radial representation) 방식을 적용하여 객체 중심점에서만 블록정합(block matching) 알고리즘으로 움직임을 추정하여 적은 계산량으로 객체 움직임을 추정함으로써 객체 윤곽선을 추적하였다. 객체의 움직임을 추정함으로써 프레임 간 객체의 움직임이 객체 중심에서 윤곽까지 거리 이상 빠르게 이동하는 객체도 추적할 수 있다. 에너지 수렴 과정에서 기울기 영상과 차영상(difference image)을 에너지 함수로 함께 사용함으로써 복잡한 배경 등에도 강건하도록 하였다. 실험 결과 움직임이 빠른 객체와 복잡한 배경 속의 객체도 실시간으로 강건하게 추적함을 확인하였다.

## Abstract

This paper proposed a object tracking algorithm which can track contour of fast moving object through motion estimation. Since the proposed tracking algorithm is based on the radial representation, the motion estimation of object can be accomplished at the center of object with the low computation complexity. The motion estimation of object makes it possible to track object which move fast more than distance from center point to contour point for each frame. In addition, by introducing both gradient image and difference image into energy functions in the process of energy convergence, object tracking is more robust to the complex background. The results of experiment show that the proposed algorithm can track fast moving object in real-time and is robust under the complex background.

**Keywords :** Fast moving object tracking, real time tracking, active rays, motion estimation

## I. 서 론

실시간 객체 추적은 최근 HCI (human computer interaction), 감시 시스템, 무인 자동차 시스템, 서비스 로봇 등의 시각 시스템을 위해서 활발히 연구되고 있다<sup>[1]</sup>. 특히 카메라로부터 실시간으로 입력되는 동영상에서 움직이는 객체의 윤곽을 추적하는 경우에는 적은 계산량으로도 객체의 움직임을 효율적으로 추정하여 이를 추적할 수 있어야 한다.

국부 최적화 방식의 그리디 스네이크 (greedy snakes)<sup>[2]</sup>를 윤곽선 추출에 적용한 객체 추적 알고리즘<sup>[3-5]</sup>은 객체 윤곽선 부근의 적은 영역 정보만을 처리하기 때문에 처리할 정보의 양이 적고, 객체와 배경을 구

분해 줄 필요가 없어서 추적 하고자 하는 객체 외에 다른 객체의 움직임에 영향을 받지 않는 장점이 있다. 그러나 객체의 윤곽선이 약할 경우에는, 윤곽선을 놓쳐 능동 윤곽 (active contour)이 수축하는 경향이 있고, 방사상 표현 (radial representation) 방식<sup>[6]</sup>을 기반으로 한 객체 추적보다 윤곽점의 탐색 영역이 많아 계산 시간이 더 요구된다.

그리디 스네이크 기반 객체 추적 알고리즘에 있어서 Lefèvre 등<sup>[4]</sup>은 기울기 영상 뿐 아니라 배경과 객체의 색상 차이를 에너지 함수에 적용하여 객체의 추적 성능을 향상시켰다. 그러나 객체 윤곽의 변형이 심한 경우에는 정확한 수렴을 보장하지 못할 뿐 아니라, 객체 이동에 대한 움직임을 추정하지 않기 때문에 현재 프레임의 객체가 이전 프레임의 수렴 결과를 완전히 벗어날 경우 객체 추적이 어려워진다.

한편, Pardàs 등<sup>[5]</sup>은 그리디 스네이크를 적용하여 윤곽 변형을 추정하여 객체를 추적하는 알고리즘을 제안하였다. 이 방법에서는 모든 윤곽점에서 움직임을 추

\* 정회원, 한국전력

(Korea Electric Power Corporation)

\*\* 학생회원, \*\*\* 평생회원, 경북대학교 전자전기컴퓨터 공학부

(School of Electrical Engineering &amp; Computer Science Kyungpook National University)

접수일자: 2004년10월6일, 수정완료일: 2004년12월8일

정하고 스네이크의 에너지 함수에 움직임 정보를 반영하여 윤곽선의 변형이 심한 객체 추적을 용이하게 하였다. 그러나 객체의 이동 속도가 클 경우에는 각 윤곽점의 움직임뿐만 아니라 객체 자체의 움직임까지도 포함할 만큼 탐색창이 커야하므로 계산량이 많아져 이동 속도가 빠른 객체 추적에는 한계가 있다.

한편, Denzler 등<sup>[6]</sup>은 그리디 스네이크의 연산 속도를 개선하기 위해 능동 윤곽의 탐색 영역을 2차원에서 1차원으로 줄인 방사상 표현 방식에 기반한 능동 방사선(active rays)을 제안하여 움직이는 객체를 추적하였다. 탐색 영역을 줄여 연산 시간을 감소시켰으나 객체의 이동에 대한 움직임을 추정하지 않기 때문에 이동 속도가 빠른 객체를 추적하는데 한계가 있다.

따라서 객체의 움직임을 빠르게 추정하여, 이동 속도가 빠른 객체의 윤곽선을 실시간 추적하는 윤곽선 기반 객체 추적에 관한 연구가 필요하다. 방사상 표현 기반을 적용하여 객체 추적 할 때 탐색 영역이 적어 연산 시간이 줄어들고 객체의 이동에 대한 움직임 추정하여 이동 속도가 빠른 객체도 추적할 수 있게 된다.

본 논문에서는 적은 계산량으로 객체 움직임을 추정함으로써 빠르게 움직이는 객체를 추적할 수 있고 복잡한 배경에 대해서도 강건한 객체 추적 알고리즘을 제안하였다. 방사상 표현 방식을 기반으로 하여 객체의 중심점에서만 블록정합(block matching) 알고리즘으로 객체의 움직임을 추정함으로써 적은 계산량으로 빠르게 움직이는 객체의 추적이 가능하게 하였다. 그리고 객체의 윤곽선 수렴 과정에서 기울기 영상뿐만 아니라 현재 프레임과 이전 프레임 간의 차영상(difference image)도 스네이크의 외부 에너지 함수에 반영함으로써 배경의 영향을 감소시켰다. 즉, 스네이크의 외부 에너지가 객체의 윤곽선에서는 큰 값을 갖고 배경에서는 약한 값을 갖도록 하여 객체 윤곽선으로의 수렴 속도를 가속화하도록 하였다. 제안한 알고리즘을 동영상에 적용하여 실험한 결과 복잡한 배경의 영상에서 손가락 윤곽과 움직임이 빠른 손가락 윤곽에 대해서 실시간 추적이 가능함을 확인하였다.

II. 움직임 추정 능동 방사선 기반 객체 추적

제안한 알고리즘에서는 그림 1과 같이 먼저 이전 프레임부터 현재 프레임의 객체의 움직임을 추정하고 이를 방사상 표현 방식에서 추정한 후 에너지 수렴을 통해 객체의 윤곽선을 추적한다.

1. 객체 중심점에서 움직임 추정

객체의 윤곽선을 추적하기 위해 방사상 표현 방식을 사용하였다. 그림 2에서 보는 것과 같이 객체의 중심에서 등각도로 해서 객체의 윤곽까지 길이들로 객체의 윤곽선을 나타낸다. 각도  $\theta$ 를 작게 할수록 윤곽점들의 수가 많아져 섬세하게 객체의 윤곽선을 나타낼 수 있으나 계산량은 많아지고 각도  $\theta$ 를 크게 하면 윤곽점들 수가 적어져 계산량은 적어지지만 객체의 윤곽선을 추출하는 해상력은 떨어진다.

각 윤곽점들은 등각도를 이루므로 중심까지 거리  $r_i$ 만으로 나타낼 수 있으며 이 거리들의 집합  $R$ 을

$$R = \{r_0, r_2, \dots, r_{K-1}\} \tag{1}$$

와 같이 정의하였고, 여기서  $K$ 는 윤곽점의 개수이다.  $R$ 은 매 프레임 윤곽점들이 객체 윤곽선으로 수렴 후에 저장되며  $r_i$ 에 의한 윤곽점 좌표  $(x_i, y_i)$ 는

$$(x_i, y_i) = (x_c + r_i \cos \theta, y_c + r_i \sin \theta) \tag{2}$$

로 정의된다. 여기서 중심점  $(x_c, y_c)$ 는 이전 프레임에서 윤곽점들이 객체 윤곽선으로 수렴 후에

$$(x_c, y_c) = \left( \frac{1}{K} \sum_{i=0}^{K-1} x_i, \frac{1}{K} \sum_{i=0}^{K-1} y_i \right) \tag{3}$$

와 같이 윤곽점들의 무게 중심을 구해 저장된다.

움직임 추정법에는 인접한 영상에서 화소 값의 연속적인 흐름으로 화소의 운동 방향과 속도를 벡터로 표현한 광류(optical flow)를 이용한 방법과 화소단위로 움직임 벡터를 추정하여 한 화소의 움직임 벡터가 다음 화소의 움직임 벡터를 추정하는 데 이용되는 화소 순환

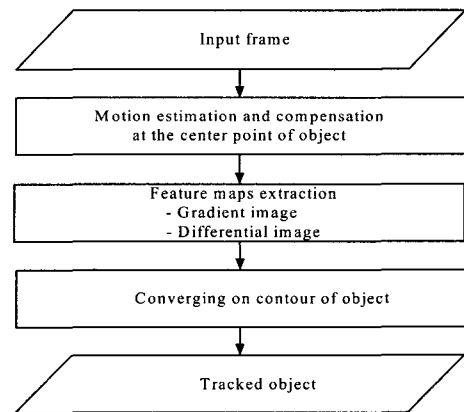


그림 1. 제안한 실시간 객체 추적 흐름도  
Fig. 1. Flowchart of the proposed real-time object tracking.

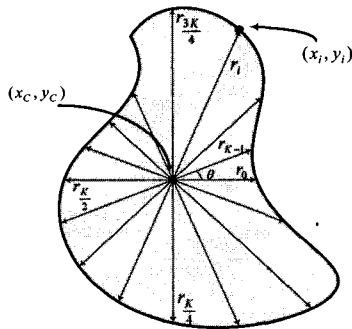


그림 2. 윤곽선의 방사상 표현  
Fig. 2. Radial representation of the contour.

알고리즘 그리고 명암, 색깔, 질감, 운동 특성, 표면 특성 등으로 입력 영상을 그 구성 요소나 객체들의 집합으로 분리하는 영역 분할 방법 등이 있다.

제안한 알고리즘에서는 전체 영상 또는 객체 영역 전체에서의 움직임 추정이 필요하지 않고 다만 객체의 중심점에서의 움직임만 추정하면 되므로, 블록정합 알고리즘을 적용하여 객체의 움직임을 추정하였다. 비교 척도로 사용되는 SAD(sum of absolute differences)는

$$SAD(u, v) = \sum_{x=0}^{B-1} \sum_{y=0}^{B-1} |I_t(x, y) - I_{t-1}(x+u, y+v)| \quad (4)$$

로 정의되며, 여기서  $I_t(x, y)$ 와  $I_{t-1}(x, y)$ 는 각각 시간  $t$ ,  $t-1$ 에서의  $(x, y)$  화소의 명도 값이고,  $(u, v)$ 는 움직임을  $B$ 는 블록 크기이며 17 [pixels]로 설정하였다. 전역 탐색 기법은 계산량이 많지만 제안한 방법에서 방사상 표현 기반을 적용하므로 객체의 중심점에서만 움직임을 추정하기 하기 때문에 적은 연산 시간으로 객체의 움직임을 추정할 수 있다.

제안한 알고리즘의 전역 탐색 기법에서는 그림 3과 같이 이전 프레임에서의 객체의 중심점을 기준으로 17 × 17 [pixels<sup>2</sup>] 크기의 매크로 블록 (macro block)을 설정하였고 현재 프레임에서의 탐색창 (search window)은 매크로 블록을 중심으로 49 × 49 [pixels<sup>2</sup>] 크기로 설정했다.

이렇게 이전 프레임의 객체 중심점에 블록정합 알고리즘을 적용하여 움직임을 추정하고, 이를 현재 프레임에서 객체 중심점으로 설정하고 중심점에서 모든 윤곽점의 거리들 집합  $R$ 을 통해 모든 윤곽점들에서 움직임이 추정하였다.

객체의 중심점에서 움직임을 추정할 때 객체가 매크로 블록 보다 크고 객체의 중심점 부근에서 화소의 명도 값이 변화가 적은 경우 움직임 추정에 오차가 수반

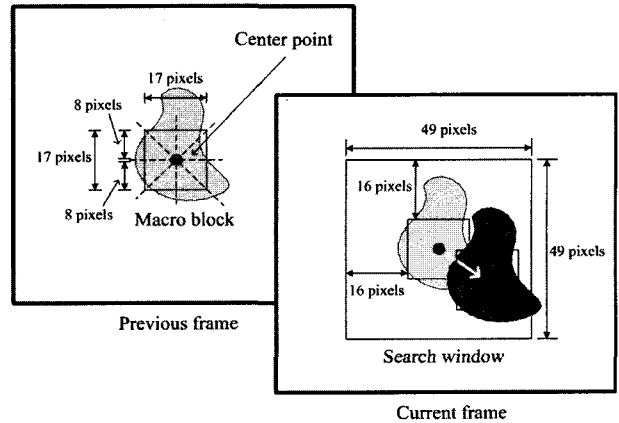


그림 3. 객체 중심점에서의 움직임 추정  
Fig. 3. Motion estimation at the center point.

될 수 있다. 이런 오차를 판단하고 보완하기 위해 매 프레임에서 추정된 객체의 움직임을 저장해 둔다. 객체의 움직임은 매 프레임 변화가 적고 객체의 움직임은 시간적으로 상관성이 높으므로 선형 예측(linear prediction)법을 도입하여 이전 프레임들의 객체 움직임으로부터 현재 프레임에서 객체 움직임을 예측한다. 프레임  $n$ 에서 움직임 예측은

$$C_n = \sum_{i=1}^P a_i c_{n-i} \quad (5)$$

로 정의되며,  $P$ 는 예측에 사용되는 이전 프레임들의 수이고  $c_{n-i} = (x_{n-i}, y_{n-i})$ ;  $i = 1, 2, \dots, P$ 는 이전 프레임들에서 객체 움직임들이며  $a_1, a_2, \dots, a_P$ 는 상대적인 가중치를 제어하는 계수이다.

이렇게 선형 예측법으로 예측한 움직임과 객체의 중심점에서 추정한 움직임 오차 범위 안에 있을 시에는 움직임 추정에 오차가 없는 것으로 판단하고, 오차 범위를 벗어 났을 경우 움직임 추정할 중심 위치는

$$\begin{cases} (x_C + r_0, y_C) & \text{if } x_{n-i} > y_{n-1} \\ (x_C, y_C + \frac{r_K}{4}) & \text{otherwise} \end{cases} \quad (6)$$

로 이동한다. 전 프레임의 움직임 성분 중  $x$ 축으로 값이 크면 움직임 추정 중심 위치를  $x$ 축으로 이동하고 그렇지 않을 경우는  $y$ 축으로 이동하여 추정한다. 여기서  $r_0$ 은 방사상 표현법에서  $\theta = 0^\circ$ 일 때 중심점에서 객체 윤곽까지 거리이며,  $\frac{r_K}{4}$ 는  $\theta = 90^\circ$ 일 때 중심점에서 윤곽까지 거리이다. 이렇게 객체의 경계에서 움직임을 추정할 경우 객체와 배경의 명도의 차가 있으므로 움직임 추정 오류를 보완할 수 있다.

2. 객체 윤곽선으로 수렴

그리디 스네이크에서 윤곽점 수렴 영역과 방사상 표현 방식에서 윤곽점의 수렴 영역은 그림 4, 5에서 보는 것과 같다. 그리디 스네이크에서는 각 윤곽점들이 탐색할 이웃 화소가 8개 인데 비해 방사상 표현 방식은 객체의 중심점에서 방사상 방향에 해당되는 영역만을 탐색하므로 탐색할 이웃 화소가 2개가 된다. 이와 같이 탐색할 영역이 그리디 스네이크 보다 적어 연산 시간을 단축시킴으로 방사상 표현 방식을 적용하여 객체의 윤곽으로 수렴하는 것이 더 효율적임을 알 수 있다. 객체를 추적하기 위한 전 단계로 초기화 과정이 있다. 기존 초기화 방법에는 분할 (segmentation) 알고리즘<sup>[7]</sup>과 수동 (user driven)<sup>[8]</sup>등이 있다. 제안한 방법에서는 먼저 미리 획득해 놓은 배경 영상과 처음 프레임의 차영상을 구하고, 열림 (opening) 연산과 닫힘 (closing) 연산으로 차영상 내의 잡음을 제거한다. 이렇게 해서 얻은 이진 영상은 객체에 대한 영역을 나타내는데 그림 6과 같이 이 영역에 대해 최소 외접 사각형을 구하고 최소 외접 사각형으로부터 객체의 중심점을 계산한다.

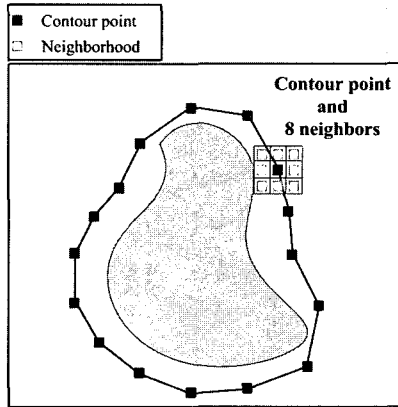


그림 4. 그리디 스네이크 탐색 영역  
Fig. 4. Search area of greedy snake.

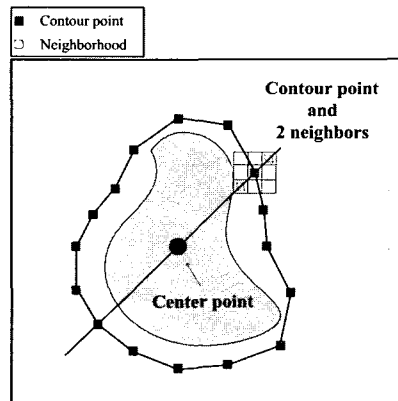


그림 5. 방사상 표현 방식 탐색 영역  
Fig. 5. Search area of radial representation.

이 때 추적하고자 한 객체 외에 또 다른 객체가 영상 내 있을 경우 최소 외접 사각형의 크기가 커지게 되어 초기화에 오차를 수반하므로 최소 외접 사각형의 크기에 제한을 두어 추적하고자 하는 객체만을 포함하는 최소 외접 사각형을 구한다. 이렇게 최소 외접 사각형으로부터 구한 중심점으로부터 등각도로 최소 외접 사각형 위로 윤곽점들을 두고 객체 윤곽까지 수렴시켜 초기화 한다.

한편 객체의 중심에서 방사상 영역으로 움직이는 윤곽 곡선은  $v(s)$ 는

$$v(s) = (x(s), y(s)), s \in [0, 1] \tag{7}$$

와 같이 정의되고, 에너지 함수  $E_{radial}$ 은

$$E_{radial} = \int_0^1 E_i(v(s)) + E_e(v(s)) ds \tag{8}$$

와 같고, 여기서  $E_i$ 는 내부 에너지를  $E_e$ 는 외부에너지 나타내며, 내부 에너지  $E_i$ 는

$$E_i(v(s)) = \frac{1}{2} (\alpha(s)|v'(s)|^2 + \beta(s)|v''(s)|^2) \tag{9}$$

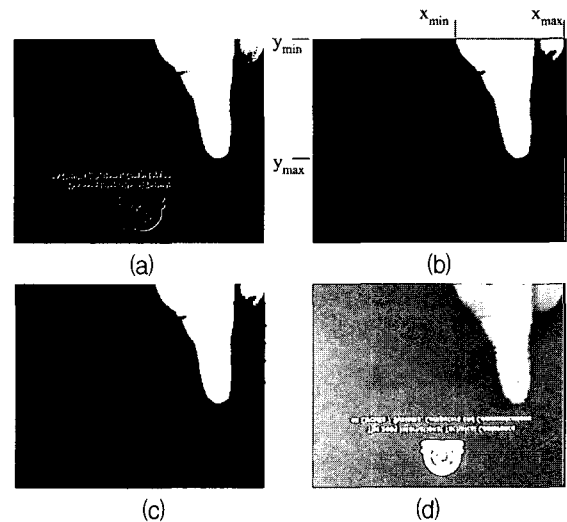


그림 6. 초기화 단계: (a) 배경 영상과 처음 프레임 영상의 차영상, (b) 열림과 닫힘 연산 후의 최소 외접 사각형, (c) 최소 외접 사각형 위의 윤곽점들 및 (d) 객체 윤곽으로 수렴 후 초기화된 윤곽점들

Fig. 6. Step of initialization: (a) Difference image of first frame and background images, (b) minimum bounded rectangle after opening and closing operation, (c) contour points on the minimum bounded rectangle, and (d) initialized contour points after converging on object contour.

와 같이 정의된다. 내부 에너지의 일차 미분항  $v'(s)$  는 이웃하는 윤곽점들 간의 거리의 균일성을 제어하고, 이차 미분항  $v''(s)$  는 윤곽선이 원의 형태와 같이 부드러운 곡선이 되도록 곡률을 제어하며  $a(s), \beta(s)$  는 상대적인 가중치를 제어하는 계수이다.

외부 에너지  $E_e$  는

$$E_e(v(s)) = -|aE_{-}(v(s)) + bE_{diff}(v(s))| \quad (10)$$

와 같이 정의되며 영상 힘 (image force)을 나타낸다. 이들 에너지의 최소화 과정을 반복 수행하여 영상 내에 관심 있는 영역의 특징인 라인, 에지 및 윤곽선 등을 추출한다.  $E_{-}(v(s))$  는 소벨 연산자 (Sobel operator)로 구한 기울기 영상의 에너지이고  $E_{diff}(v(s))$  는 현재 프레임과 이전 프레임의 명도 차이로 구한 차영상의 에너지이며,  $a, b$  는 이들의 상대적인 가중치를 제어하는 계수이다. 그림 7은 에너지 함수에 사용되는 기울기 영상과 차영상의 예를 보여준다. 이렇게 차영상을 기울기 영상과 함께 에너지 함수로 사용함으로써 움직이는 객체의 윤곽을 더 강화하고 복잡한 배경의 영향은 감소시켰다.

### 3. 방사상 표현 방식에서의 움직임 추정의 장점

객체의 움직임 추정 없이 에너지 수렴 과정으로만 객체를 추적할 경우 움직임이 큰 객체 추적이 어려워진다. 그림 8과 같이 움직임 추정 없이 에너지 수렴 과정으로만 객체 추적 경우 추적할 수 있는 객체 움직임의 한계가 움직임 벡터의 크기와 객체 중심점에서 윤곽점들로 거리 중 움직임 벡터의 역방향 성분의 거리의 관계로 정해진다.

움직임 추정 없이 객체 추적하는 경우 추적의 한계는 움직임 벡터 크기  $|\vec{m}|$ 와 움직임 방향으로의 객체 중심에서 윤곽까지 거리들 중 역방향의 거리가  $r_r$ 의 관계가

$$|\vec{m}| > r_r \quad (11)$$

와 같고, 이와 같은 경우는 이전에 추적된 객체의 윤곽점에서부터 현재 프레임의 객체 윤곽까지 수렴할 수 있는 범위를 벗어나게 되어 객체를 추적할 수 없게 된다. 그림 9는 움직임 벡터 크기가 객체 중심점에서 객체 윤곽까지 거리보다 클 경우의 실험 결과이다.

움직임 추정이 없는 기존 방사상 표현 방식에서 객체 추적의 한계는 이론적으로 움직임 벡터 크기가 객체 중

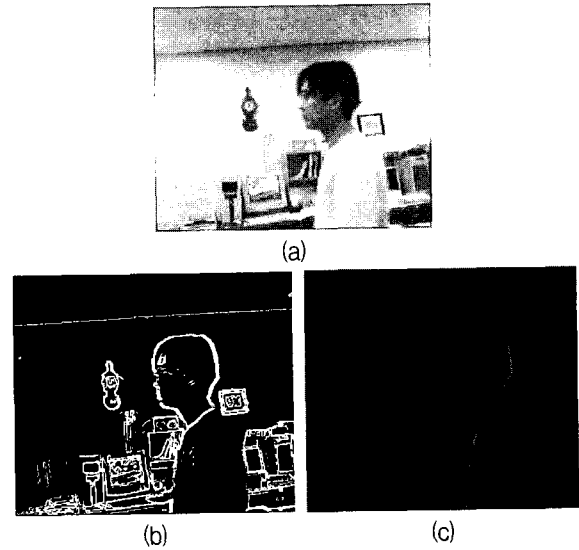


그림 7. 얼굴 추적 실험에서 에너지 함수에 사용되는 영상들: (a)원영상, (b)기울기 영상 및 (c)차영상  
Fig. 7. Images used as energy function at the face tracking experiment: (a) Original image, (b) gradient image, and (c) difference image.

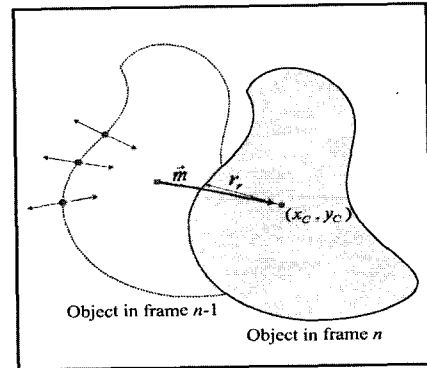


그림 8. 움직임 추정 없는 객체 추적  
Fig. 8. Object tracking without motion estimation.

심점에서 객체 윤곽까지 거리보다 클 경우이지만 실제 적용에 있어서는 윤곽점의 탐색 영역을 크게 하더라도 내부 에너지 함수와 객체 종류 등이 원인이 되어 움직임 벡터 크기가 객체 중심점에서 객체 윤곽까지 거리보다 작더라도 추적할 수 없는 경우가 있다. 그러나 그림 10과 같이 에너지 수렴 과정 전에 객체의 움직임을 추정해 주는 단계를 수행하면 객체의 움직임이 큰 객체의 경우도 추적할 수 있게 된다. 따라서 제안한 방법에서는 객체의 최대 움직임 크기는 움직임 추정의 탐색창 크기에 한계가 정해진다.

움직임 추정이 없이 객체 추적하는 기존 방법에서 추적 가능한 한 프레임 사이에서의 객체 최대 움직임 크기  $d_{max\_c}$ 는

$$d_{max\_c} = (r_r \pm \Delta r_r) - c \quad (12)$$

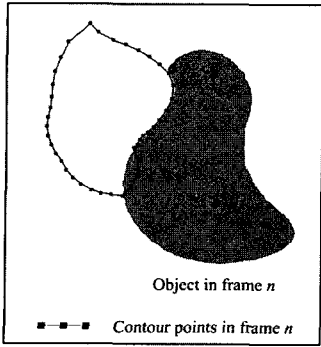


그림 9. 움직임 추정 없는 객체 추적 실험 결과  
 Fig. 9. Result of object tracking without motion estimation.

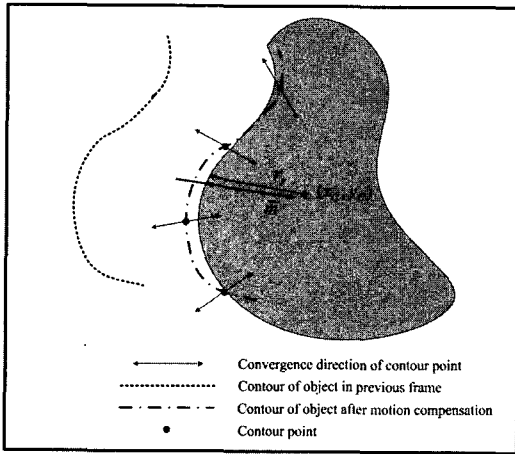


그림 10. 움직임 추정 후 객체 추적  
 Fig. 10. Object tracking after motion estimation.

이와 같이 정의된다. 여기서  $c$ 는 객체 윤곽선의 특성과 윤곽점들의 균일성과 곡률을 제어하는 일차 미분항과 이차 미분항에 의해서 결정되는 수렴 범위에 의한 성능 저하 인자이다.  $\Delta r_r$ 은 비강성 객체의 경우 프레임 간 객체 중심에서의  $r_r$ 의 변화량이다. 그리고 제안한 방법에서 추적 가능한 한 프레임 사이에서의 객체 최대 움직임 크기  $d_{max\_p}$ 는

$$d_{max\_p} = L_s / 2 - L_m / 2 \quad (13)$$

이와 같이 정의되고  $L_s$ 는 탐색창의 한 번의 길이이고,  $L_m$ 은 매크로 블록의 크기이다. 한편 제안한 방법에서는  $\Delta r_r$ 과  $c$ 는 추적 가능 여부가 아닌 수렴의 질에 영향을 미치게 되며

$$\Delta r_r < c \quad (14)$$

일 경우에는 안정적인 수렴특성을 보장할 수 있다. 따

라서 기존 방법의  $d_{max\_c}$ 는 객체의 크기와 속도에 의해서 결정되고 제안한 방법의  $d_{max\_p}$ 는 매크로 블록의 크기에 영향을 받는다.

### III. 실험결과 및 고찰

실험은 1.7GHz Pentium 4 PC 상에서 수행하였으며, 웹캠 (webcam)으로 획득한 352×288 크기의 동영상 (그림 11, 12)과 176×144 크기의 동영상 (그림 13, 14)에 적용하였다. 그림 11과 12는 가변적인 이동 속도로 움직이는 객체를 포함하고 있으며, 그림 13과 14는 각각 8 [pixels/frame]과 16 [pixels/frame]의 등속 운동을 하는 객체를 포함하고 있다.

먼저, 이동과 회전을 같이 하는 객체의 추적 실험을 위해서 그림 11과 같이 실내에서 이동하면서 회전하는 사람의 머리를 포함하는 동영상에 적용하여 성공적으로 윤곽을 추적함을 확인 하였다.

복잡한 배경 하에서 비강성 객체 추적 실험을 위해 그림 12와 같이 복잡한 배경 하에서 관절을 굽히는 운동을 하는 손가락 윤곽을 추적하였다. 손가락의 모양이 고정되어 단순하게 움직이지 않고 초기 손가락이 펴진 상태의 직선 모양에서 곡선 모양으로 손가락을 굽힘으로 비강성 객체 추적 실험을 하였다. 객체의 배경은 여러 글자들이 나열되어 있는 책으로 하였다. 이 실험 영상의 기울기 영상을 추출하여 볼 때 글자들로 인한 기울기가 커서 글자들이 객체에 대해 복잡한 배경이 된다. 그림 12에서 보는 바와 같이 글자로 인한 복잡한 배경에도 비강성 객체 손가락 윤곽을 추적함을 보여준다.

그리고 움직임이 빠른 객체 실험을 위해 30 [frames/sec]의 동영상에서 그림 13과 같이 프레임 간 8 화소씩 이동하여 240 [pixels/sec] 속도로 움직이는 손가락 윤곽 추적 실험과 그림 14와 같이 프레임 간 16 화소씩 이동하여 480 [pixels/sec] 로 움직이는 손가락 윤곽을 추적 실험하였다. 그림 13과 14는 240 [pixels/sec] 속도와 480 [pixels/sec] 속도로 움직이는 손가락 윤곽을 성공적으로 추적함을 보여준다. 이 실험에서 사용한 파라미터들은 표 1과 같다.

표 2는 Denzler 등<sup>[6]</sup>이 제안한 알고리즘의 추적 가능한 객체의 최대 속도와 제안한 알고리즘 객체 최대 속도 비교표이다. Denzler 등<sup>[6]</sup>이 제안한 알고리즘에서는 최대 움직임 벡터의 크기가 8 [pixels/frame]이므로 프레임 속도가 30 [frames/sec]일 때 추적 가능한 객체의

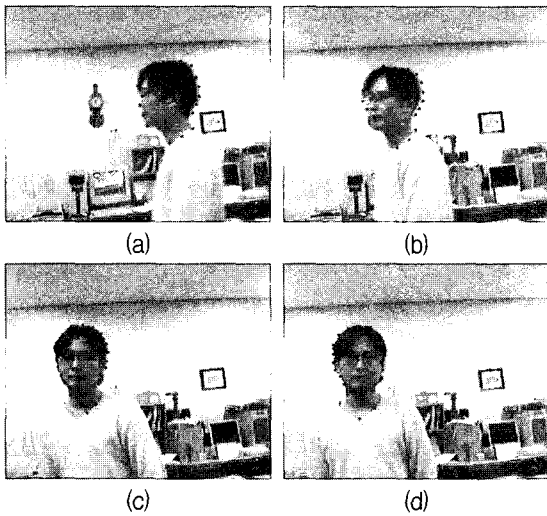


그림 11. 머리 윤곽 추적  
Fig. 11. Head-contour tracking: (a) frame 5, (b) frame 15, (c) frame 25, and (d) frame 35.

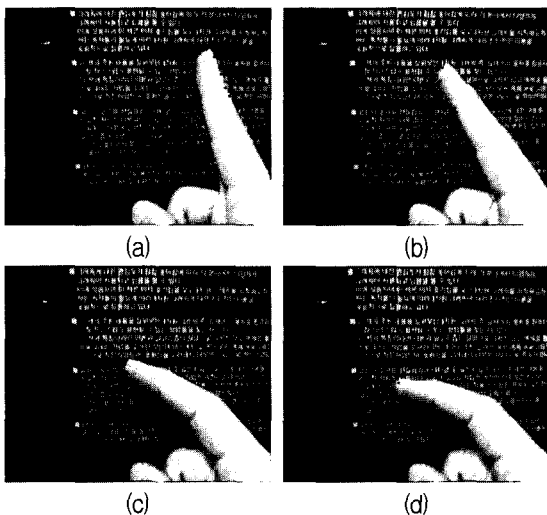


그림 12. 비강성 객체 손가락 윤곽 추적  
Fig. 12. non-rigid finger-contour tracking: (a) frame 5, (b) frame 15, (c) frame 25, and (d) frame 35.

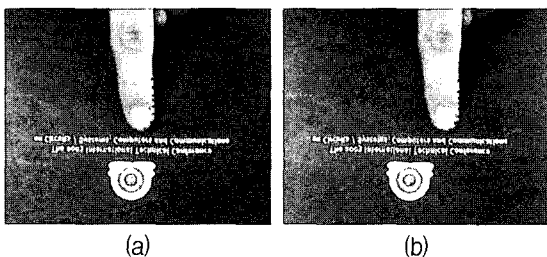


그림 13. 손가락 윤곽 추적 (240 [화소/초])  
Fig. 13. Finger-contour tracking (240 [pixels/sec]): (a) frame 20 and (b) frame 26.

최대 속도는 240 [pixels/sec]가 되고 제안한 알고리즘에서는 최대 움직임 벡터의 크기가 16 [pixels/frame]이므로 프레임 속도가 30 [frames/sec]일 때 추적 가능한

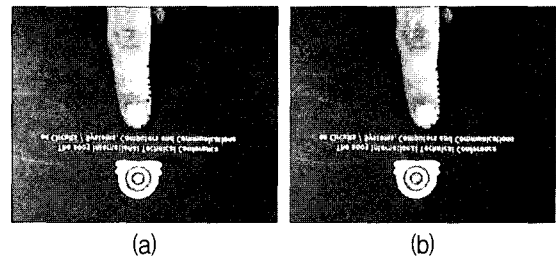


그림 14. 빠르게 움직이는 손가락 윤곽 추적 (480 [화소/초])  
Fig. 14. Fast moving finger-contour tracking (480 [pixels/sec]): (a) frame 12 and (b) frame 13.

표 1. 그림 13과 14 실험에서 사용한 파라미터들  
Table 1. Parameters used in the experiments in Fig. 13 and Fig. 14.

Parameter	Value
Frame rate ( $F_r$ )	30 [frame/sec]
Search window size ( $(L_s - L_m) / 2$ )	16 [pixels]
Image resolution	176×144 [pixels <sup>2</sup> ]
Distance from center point to contour point in the same direction of object's movement ( $r_r$ )	12 [pixels]

표 2. 추적 가능한 객체의 최대 속도 비교  
Table 2. The results of maximum velocity of object.

	Denzler et al. <sup>[8]</sup>	Proposed algorithm
Maximum size of motion vector	8 [pixels/frame]	16[pixels/frame]
Maximum velocity of object to be tracked	240 [pixels/sec]	480[pixels/sec]

표 3. 그림 11 실험에서 프레임 당 객체 추적 처리 시간 비교  
Table 3. Comparison of the processing time in a frame for the experiment of Fig. 11.

[Pentium IV, 1.7 GHz]

Algorithm	Processing time [msec]
Conventional <sup>[4]</sup>	15.1
Proposed	14.9

객체의 최대 속도는 480 [pixels/sec]가 된다. 식 (12)에 따라 수렴 범위에 의한 성능 저하 인자  $c$ 가 0일 경우에 Denzler 등이 제안한 방법은 최대 움직임 벡터의 크기가 12 [pixels/frame]이 되지만  $c$ 의 영향에 의해 실험에서 8 [pixels/frame]이 되었다. 이를 30 [frames/sec]의 동영상에서 초당 이동 거리로 환산하면 기존 방법은 240 [pixels/sec]이지만 제안한 방법은 480 [pixels/sec]가 된다. 따라서 제안한 알고리즘이 기존의 방법보다 움직임이 더 빠른 객체도 추적할 수 있음을 말해준다.

표 3는 기존의 그리디 스네이크를 객체 추적에 적용한 경우와 제안한 방법으로 객체를 추적했을 경우 프레임 당 평균 시간을 비교하였다. 제안한 방법이 움직임을 추정함에도 불구하고 중심점에서 한 번 수행하여 추정하고 방사상 표현 기반으로 하여 탐색 영역을 줄임으로 기존 그리디 스네이크를 적용한 객체 추적 알고리즘의 처리속도에 큰 차이가 없음을 확인하였다.

제안한 알고리즘에서 객체 움직임을 객체 중심점에서 블록정합 알고리즘을 적용하여 17 × 17 [pixels<sup>2</sup>] 크기의 매크로 블록을 설정하고 추정하였으나 객체의 크기와 모양을 고려하지 않아 객체 종류에 따라 움직임 추정에 오차를 수반할 수 있다. 따라서 매크로 블록을 설정할 때 객체의 크기와 모양을 고려한 적응적인 매크로 블록 설정의 연구가 추후 연구 대상이다.

#### IV. 결 론

본 논문에서는 방사상 표현 방식을 적용하여 객체의 중심점에서 움직임을 추정하고 객체의 윤곽선을 추적하는 알고리즘을 제안하였다. 방사상 표현 기반 객체 추적을 적용함으로써 객체의 움직임을 추정하는데 객체 중심에서 블록정합 알고리즘을 한번 적용하여 움직임을 추정하였고, 객체의 중심점에 움직임을 추정하는 것이 객체의 전 영역에 움직임을 추정하는 효과를 얻게 되어 움직임 추정에 연산 시간을 줄였다. 이렇게 객체의 윤곽선 수렴에 객체의 객체 움직임을 추정하여 프레임 당 객체 중심에서 윤곽까지 거리보다도 움직임이 큰 객체를 추적 할 수 있었다. 스네이크 외부 에너지 함수로 기울기 영상과 함께 차영상을 사용하여 배경의 영향을 감소시키므로 복잡한 배경에서도 강건하게 객체를 추적하게 하였다.

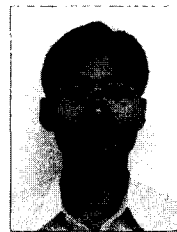
복잡한 배경의 영상 실험과 움직임의 속도가 빠른 객체 실험에서 강건하게 객체를 추적함을 확인하였다. 제안된 알고리즘은 얼굴 인식의 전 단계에서 사람의 머리를 추적하여 얼굴 영상을 획득하는 것이나 HCI에 적용되기 위한 사람의 손가락을 추적하는데 이용될 수 있다.

#### 참 고 문 헌

[1] A. L. Gilbert, "A Real-Time Video Tracking System," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 47-56, January 1980.  
 [2] D. J. Williams and M. Shah, "A Fast Algorithm

for Active Contours and Curvature Estimation," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 55, pp. 14-26, January 1992.  
 [3] C. L. Lam and S. Y. Yuen, "An Unbiased Active Contour Algorithm for Object Tracking," *Pattern Recognition Letters*, vol 19, pp. 491-498, April 1998.  
 [4] S. Lefèvre, J. G. Gerard, A. Piron, and N. Vincent, "An Extended Snake Model for Real-Time Multiple Object Tracking," *RFAI publication: International Workshop on Advanced Concepts for Intelligent Vision Systems*, Ghent Belgium, pp. 268-275, September 2002.  
 [5] M. Pardàs and E. Sayrol, "Motion Estimation Based Tracking of Active Contours," *Pattern Recognition Letters*, vol. 22, pp. 1447-1456, November 2001.  
 [6] J. Denzler and H. Niemann, "Active Rays: Polar-Transformed Active Contours for Real-Time Contour Tracking," *Real-Time Imaging 5*, Singapore, pp. 203-213, 1999.  
 [7] M. Takahata, M. Imai, and S. Tsuji, "Determining Motion of Non-Rigid Objects by Active Tubes," *International Conference on Pattern Recognition*, pp. 647-650, 1992.  
 [8] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 294-302, March 1995.

#### 저 자 소 개



라 정 중(정회원)  
 1999년 경북대학교 전자공학과  
 학사 졸업.  
 2004년 경북대학교 전자공학과  
 석사 졸업.  
 <주관심분야 : 영상처리, 병렬분산  
 처리, 임베디드 프로세서>

서 경 석(학생회원)  
 제38권 SP편 3호 참조

최 흥 문(평생회원)  
 제38권 SP편 3호 참조