

논문 2011-48SP-6-12

# 개선된 스네이크 알고리즘을 이용한 객체 윤곽 추적

## ( Object Contour Tracking Using an Improved Snake Algorithm )

김진율\*, 정재기\*\*

( Jinyul Kim and Jaeki Jeong )

### 요 약

스네이크 알고리즘은 객체와 배경 사이의 활성 윤곽(active contour)을 추출하여 객체를 추적하는 기법으로 널리 사용되고 있지만, 객체 윤곽의 밝기 기울기 성분보다 배경에 존재하는 기울기 성분이 크면 객체 윤곽이 배경으로 잘못 수렴되는 문제를 갖는다. 또 객체의 급격한 이동으로 인해 객체의 윤곽이 탐색영역을 벗어나면 윤곽선이 객체의 내부로 수축되는(shrink) 현상이 발생하게 되어 객체 추적이 실패하게 된다. 본 논문에서는 이러한 기존의 문제점을 개선한 새로운 스네이크 윤곽 추적 방법을 제안한다. 먼저, 객체 경계에 존재하는 평균 기울기 방향만을 고려하도록 개선된 에지 에너지 함수와 스플라인 경계의 안쪽과 바깥쪽 영역의 명암차를 이용한 컨트라스트 에너지 함수를 제안하여 윤곽선이 배경에 잘못 수렴되는 문제를 해결하였다. 또한 이전 프레임과 현재 프레임의 차영상으로부터 스네이크 포인트의 모션 벡터를 얻고 이를 이용하여 이전 프레임의 스네이크 포인트를 현재 프레임의 객체 윤곽 부근으로 빠르게 이동시켜 윤곽선이 객체 내부로 수축되는 현상을 방지하였다. 실험 결과 제안하는 기법은 기존 방법들에 비하여 복잡한 배경에 더 강인하며 움직임이 큰 객체를 정확하게 추적할 수 있었다.

### Abstract

The snake algorithm is widely adopted to track objects by extracting the active contour of the object from background. However, it fails to track the target converging to the background if there exists background whose gradient is greater than that of the pixels on the contour. Also, the contour may shrink when the target moves fast and the snake algorithm misses the boundary of the object in its searching window. To alleviate these problems, we propose an improved algorithm that can track object contour more robustly. Firstly, we propose two external energy functions, the edge energy and the contrast energy. One is designed to give more weight to the gradient on the boundary and the other to reflect the contrast difference between the object and background. Secondly, by computing the motion vector of the contour from the difference of the two consecutive frames, we can move the snake pointers of the previous frame near the region where the object boundary is probable at the current frame. Computer experiments show that the proposed method is more robust to the complicated background than the previously known methods and can track the object with fast movement.

**Keywords :** object tracking, active contour, snake algorithm, energy function, motion vector

\* 정회원-교신저자, 수원대학교 전자공학과  
(Dept. of Electronic Engineering,  
Suwon University)

\*\* 정회원, UDworks(주)  
(UDworks Co. Ltd.)

※ 이 논문은 2008년도 정부재원(교육인적자원부 학술 연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2008-521-D00298). 그리고 경기도의 경기도지역협력연구센터(GRRC) 사업 [(GRRC 수원2011-B2), U-city 보안감시 기술협력 센터]의 일환으로 수행하였음.

접수일자: 2011년6월17일, 수정완료일: 2011년10월11일

## I. 서 론

영상 추적(tracking)은 연속되는 영상 프레임 내에서 움직이는 사람의 얼굴과 같은 특정한 객체를 감지하여 추적하는 기술로 최근 보안 감시 시스템, 로봇비전, 화상통화, HCI(Human Computer Interaction) 등의 응용 분야에 폭 넓게 이용되고 있다. 영상 추적이 다양한 응용 분야에 적용되기 위해서는 복잡한 배경이 존재하는 영상에서 다양한 객체의 움직임을 정확하고 안정적으로

추적할 수 있는 정교한 알고리즘이 요구되고 있으며, 이를 만족하기 위하여 많은 연구들이 있어왔다.

영상추적에는 객체 내부의 색상에 근거하여 추적하는 방법으로 Mean Shift<sup>[1]</sup>, 색상 기반 Particle Filter<sup>[2~4]</sup> 등을 들 수 있다. Mean Shift는 반복 연산을 통해 추적 객체와 가장 유사한 색상 분포를 갖는 영역을 추적하며, 색상 기반 Particle Filter는 다수의 샘플을 통하여 영상 내의 추적 객체 색상과 유사한 확률 분포를 추정하여 추적한다. 한편, 객체의 윤곽선을 추정하는 추적 방법으로 ACM(Active Contour Model) 기반 방법<sup>[5~11]</sup>들이 있다. ACM 기반 방법들은 객체의 윤곽선을 표현하는 포인터들이 움직이는 객체의 윤곽선을 지속적으로 따라 가도록 제어하여 배경과 객체를 분할(segmentation)함으로써 객체를 추적한다. 객체 색상을 사용하는 알고리즘은 객체의 움직임이 크거나 모션 블러가 생겨도 색상분포의 변화가 적으므로 비교적 강한 추적 성능을 보이지만 객체 형태나 변형을 정확하게 추정할 수 없다. 반면에 ACM 기반 방법들은 객체의 형태 변형을 추적할 수 있지만 객체가 급격하게 움직이거나 배경이 복잡하면 추적의 정확도가 떨어진다. 본 논문에서는 객체의 형태 변형을 정확하게 추적하기 위하여 복잡한 배경이나 객체의 급격한 움직임에도 대응할 수 있도록 기존의 ACM 기법들의 단점을 개선한 새로운 객체 추적을 제안한다.

Kass<sup>[5]</sup>에 의해 제안된 ACM 기법은 스네이크 알고리즘이라고 불리며 객체를 배경으로부터 분할하는 방법이다. 스네이크 알고리즘은 객체의 윤곽을 빠르게 추출하지만 배경에 존재하는 강한 기울기 성분에 잘 못 수렴할 수 있고 모션 블러나 객체의 급격한 움직임이 있는 경우 윤곽선이 객체의 내부로 수축(shrink)되는 현상이 발생하는 문제점이 있다. 이러한 문제점을 해결하기 위하여 다양한 방법들이 제안 되었다. Xu<sup>[8]</sup>는 기울기의 흐름을 생성하여 스네이크 포인터가 윤곽에 도달 할 수 있도록 만드는 GVF (Gradient Vector Flow) 기법을 제안하였으며, Leymarie<sup>[9]</sup>은 다중 해상도를 이용하여 넓은 지역을 탐색 후 국부 지역에 수렴하도록 HDC (Hierarchical Discrete Correlation Method) 기법을 제안하였다. 또한 Lee<sup>[10]</sup>는 이전과 현재 프레임의 합성 기울기를 통하여 배경을 제거하는 기법을 제안하였다. 위의 기법들은 복잡한 배경과 객체 윤곽에 존재하는 기울기 성분을 완벽하게 구분하지 못하여 배경에 존재하는 기울기 성분에 수렴하게 되는 문제점이 있다. 또한 넓

은 지역의 탐색은 객체 내부에 강한 기울기 성분이 존재 하면, 객체 내부에 수렴하여 객체 추적에 실패의 요인으로 작용하게 된다. 최근 이러한 문제점을 개선하기 위해 김<sup>[11]</sup>은 이진 차별 마스크를 적용한 이전과 현재 프레임의 두 단계 합성 기울기 (Two-level Composite Gradient: 이하 TCG) 맵 기법을 제안하였다. TCG 맵 기법은 큰 움직임이 존재하는 객체 추적에서 기존 GVF, HDC 기법들 보다 추적 성능이 우수하지만 차별 마스크 성능에 따라 추적 성능이 의존되게 된다.

본 논문에는 복잡한 배경 잡음이 존재할 때도 윤곽선이 배경으로 수렴하지 않도록 하는 두 가지 외부 에너지 함수를 제안하였다. 또, 객체가 급격히 움직이는 경우에도 스네이크 포인터를 객체의 윤곽선 근처로 빠르게 이동시켜 객체 윤곽선이 수축되는 현상을 막아 추적 성능을 높이는 모션 벡터 계산법을 제안하였다.

본 논문의 구성은 다음과 같다. II장에서는 기존 스네이크 알고리즘을 기술하고 객체 추적에서의 문제점을 보인다. III장에서는 제안하는 개선된 외부 에너지 함수와 차영상 기반의 모션 벡터에 대하여 기술한다. IV장에서는 기존의 기법들과 제안하는 방법을 컴퓨터 실험을 통해 비교하여 제안된 방법이 기존의 스네이크 기반 알고리즘들 보다 우수한 객체 추적 성능을 나타냄을 보인다. 마지막으로 V장에서는 결론을 기술하였다.

## II. 스네이크 알고리즘을 이용한 추적과 문제점

스네이크 알고리즘<sup>[5, 12]</sup>을 이용하여 이전 프레임의 객체 윤곽에 존재하는 스네이크 포인터들을 현재 프레임의 객체 윤곽으로 이동시켜 객체를 추적하는 예를 그림 1에 나타내었다. 이전 프레임의 각 스네이크 포인터는 현재 좌표를 기반으로 설정된 탐색 윈도우 영역 내에서 에너지가 최소로 되는 픽셀로 이동하여 현재 프레임의 객체 윤곽을 결정하는 새로운 스네이크 포인터를 구성한다. 그림에서  $v(s) = (x(s), y(s))$ 는  $s$ 번째 스네이크 포인터의 좌표를  $w^s(i)$ 는 스네이크 포인터  $s$ 에서 정의된 탐색윈도우 내  $i$ 번째 픽셀의 좌표를 각각 나타낸다.

스네이크 포인터들을 객체 윤곽으로 수렴하도록 하는 에너지 함수  $E_{snake}$ 는 다음 식과 같이 정의<sup>[5]</sup> 된다.

$$E_{snake} = \int_0^1 \{E_{int}(v(s)) + E_{ext}(v(s))\} ds \quad (1)$$

여기서  $E_{int}$ 는 스네이크 포인터들의 관계를 결정하

는 내부 에너지를 나타내고,  $E_{ext}$ 는 스네이크 포인터를 객체의 윤곽선으로 끌어당기는 역할을 하는 외부 에너지이다. 내부에너지는 포인터간의 인력을 나타내는 수축 에너지와 인근 스네이크 포인터들의 휨 정도를 나타내는 곡률 에너지로 정의되고 외부에너지는 배경과 객체의 경계를 나타낼 수 있는 밝기 값의 기울기 정보를 이용한 에지 에너지로 정의된다.

그러나 기존 스네이크 알고리즘은 밝기의 기울기 성분에 의존하는 에지 에너지에 의해 객체의 윤곽선 기울기 성분보다 강한 기울기 성분을 갖는 배경이 존재할 경우 윤곽선이 배경으로 잘못 수렴한다. 또 객체의 이동에 의한 모션 블러 현상 등으로 윤곽선이 불확실해지

거나 객체의 빠른 움직임으로 객체의 윤곽에 탐색 영역이 도달하지 못하는 경우 윤곽선이 객체 내부로 수축되어 추적에 실패하는 문제가 존재한다.

이러한 문제점들은 그림 2에 보인 기존 스네이크 알고리즘을 이용한 얼굴 윤곽 추적 실험을 통해 잘 드러난다. 그림 2는 3가지 실험 영상에 대하여 기존 스네이크 알고리즘을 적용한 것으로 (a), (b), (c)의 경우 모두 추적하려는 얼굴 윤곽선이 강한 기울기 성분을 갖는 주변 배경으로 잘못 수렴하고 있음을 볼 수 있다. 또 (b)와 (c)의 경우에는 얼굴의 빠른 움직임으로 탐색 영역이 얼굴 경계에 도달하지 못해 윤곽이 얼굴 내부로 수축되어 객체 추적에 실패하는 현상도 함께 관측된다.

### III. 제안하는 개선된 스네이크 알고리즘

기존 스네이크 알고리즘의 문제를 해결하기 위하여 본 논문에서는 배경에 강인한 두 가지 외부 에너지 함수를 제안하고 또 스네이크 포인터를 윤곽선이 존재하는 영역으로 빠르게 이동시켜 주는 모션 벡터 사용기법을 제안한다.

본 논문에서 제안하는 개선된 스네이크 알고리즘의 전체 과정은 다음과 같다. 먼저, 이전 프레임에서 추적 객체의 윤곽선을 표현하고 있는 스네이크 포인터들이 주어졌다고 하자.

1) 이전프레임과 현재프레임의 차영상을 구하고 식 (12)로 정의되는 모션 벡터를 계산하여 스네이크 포인터를 현재 프레임에서 객체의 윤곽선이 존재할 가능성이 높은 곳으로 먼저 이동시킨다.

2) 각 스네이크 포인터에 대하여 식 (5)로 정의되는 방향성분을 고려한 에지 에너지와 식(11)로 정의되는 컨트라스트 에너지를 구하여 외부 에너지를 얻는다.

3) 에너지의 합이 최소가 되는 탐색윈도우 내 픽셀 위치로 스네이크 포인터를 이동시킨다.

제안하는 에너지 함수와 모션 벡터를 구하는 방법을 아래에 자세히 설명한다. 그림 3에 제안하는 스네이크 알고리즘을 상세히 설명하기 위해 필요한 탐색윈도우와 서브블록윈도우를 정의하였다. 주어진 스네이크 포인터  $s$ 에 대하여 일정 크기의 탐색윈도우가 설정되고 이 탐색윈도우 내의  $i$ 번째 픽셀의 좌표는  $w^s(i)$ 이다. 또 탐색윈도우내의  $i$ 번째 픽셀에 대하여 이 픽셀을 중심으로

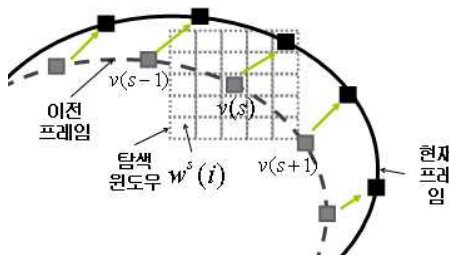


그림 1. 기존 스네이크 포인터의 이동 예  
Fig. 1. Example of the conventional snake pointer movement.



그림 2. 기존 스네이크 알고리즘을 사용한 얼굴 추적 결과  
Fig. 2. Face tracking results of the conventional snake algorithm.

하는 서브블록윈도우를 정의한다. 이 서브블록윈도우 내의  $j$ 번째 픽셀의 좌표는  $b^{s,i}(j)$ 이다.

1. 개선된 스네이크 외부 에너지 함수

본 논문에서는 기존 외부 에너지 함수 대신 다음 식으로 정의되는 새로운 외부 에너지 함수를 제안하여 사용한다.

$$E_{ext}^s(w^s(i)) = \alpha E_{edge}^s(w^s(i)) + \beta E_{contrast}^s(w^s(i)) \tag{2}$$

여기서  $E_{edge}^s(w^s(i))$ 는 윤곽선의 방향 성분을 고려한 에지 에너지이고  $E_{contrast}^s(w^s(i))$ 는 탐색 윈도우 영역 내 서브블록의 명암차를 이용하여 계산하는 컨트라스트 에너지를 나타낸다. 또,  $\alpha$ 와  $\beta$ 은 상수이다.

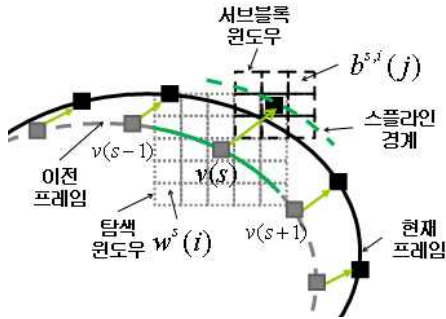


그림 3. 제안하는 스네이크 포인터의 이동 예  
Fig. 3. Example of the proposed snake pointer movement.

가. 방향 성분을 고려한 에지 에너지

기존 스네이크 알고리즘에서 에지 에너지는 밝기 값의 기울기 성분을 사용하므로 배경에 객체의 윤곽선 기울기보다 강한 기울기 성분이 존재하면 스네이크 포인터가 배경으로 수렴되는 문제가 있다. 이러한 문제점은 배경에 존재하는 기울기 성분 중 스플라인의 경계 방향과 일치하는 방향 성분만을 고려하도록 에너지 함수를 수정함으로써 크게 줄일 수 있다. 이를 위하여 스플라인 경계 위에 존재하는 기울기 성분과 탐색 윈도우 영역에 존재하는 기울기 성분을 내적하여 배경과 객체 경계의 기울기 성분을 구분할 수 있는 방향 성분을 고려한 에지 에너지를 제안한다.

구체적으로 먼저 이전 프레임에서 구한  $s$ 번째 스네이크 포인터에 대해  $s-1, s, s+1$ 번째 스네이크 포

인터를 잇는 스플라인 경계를 생성한다. 다음으로, 식 (3)과 (4)를 사용하여 이 스플라인 경계 위에 존재하는 탐색윈도우 내 픽셀들이 갖는 평균 기울기  $\mu Gradient X$ ,  $\mu Gradient Y$ 를 계산한다. 식에서  $G_X(w^s(i))$ 와  $G_Y(w^s(i))$ 는 각각 탐색윈도우 내 픽셀의 x-방향과 y-방향 기울기 성분을 나타낸다. 마지막으로 식 (5)를 사용하여 이 스플라인 경계위의 평균 기울기를 탐색 윈도우 내의 각 픽셀의 기울기 성분 ( $G_X(w^s(i))$ 와  $G_Y(w^s(i))$ )과 내적을 하여 윤곽선의 방향과 동일한 방향의 기울기만을 고려하는 에지 에너지  $E_{edge}^s$ 를 얻는다. 여기서  $f_{boundary}(w^s(i))$ 는 탐색 윈도우내의 어떤 픽셀이 스플라인 경계 위에 존재하는 픽셀인지 검사하는 함수로 식 (6)과 같이 정의된다.

$$\mu Gradient X = \frac{\sum_{i=1}^N G_X(w^s(i)) f_{boundary}(w^s(i))}{\sum_{i=1}^N f_{boundary}(w^s(i))} \tag{3}$$

$$\mu Gradient Y = \frac{\sum_{i=1}^N G_Y(w^s(i)) f_{boundary}(w^s(i))}{\sum_{i=1}^N f_{boundary}(w^s(i))} \tag{4}$$

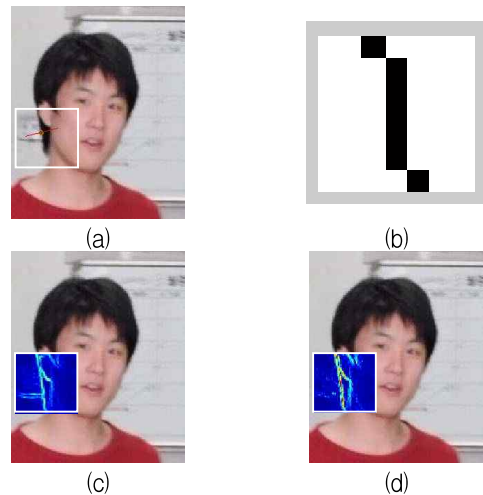


그림 4. 에지 에너지 생성 예 (a)스네이크 포인터, (b)스플라인 경계, (c)기울기 맵, (d)제안하는 에지 에너지 맵  
Fig. 4. An example of edge energy generation (a)snake pointer, (b)spline boundary, (c)gradient map, (d)proposed edge energy map.

$$E_{edge}^s(w^s(i)) = - \left| [G_X(w^s(i)) \ G_Y(w^s(i))] \begin{bmatrix} \mu_{GradientX} \\ \mu_{GradientY} \end{bmatrix} \right| \quad (5)$$

$$f_{boundary}(w^s(i)) = \begin{cases} 1, & w^s(i) \in Boundary \\ 0, & w^s(i) \notin Boundary \end{cases} \quad (6)$$

그림 4는 제안된 에지 에너지 생성 예이다. 그림 4의 (a)와 같이 사각형 영역의 중심부에 스네이크 포인터가 존재 할 때, (b)는 탐색윈도우 내 이전 프레임의 객체 윤곽을 나타내는 스플라인 경계이며, (c)는 현재 프레임에서 밝기 값의 기울기 성분 맵을 나타낸다. (d)는 방향 성분을 고려한 에지 에너지  $E_{edge}^s$  의 맵을 나타낸다. 제안하는 에지 에너지는 객체의 윤곽선에 존재하는 기울기 성분을 강화하며, 경계의 에너지 성분과 다른 기울기 성분은 약화시켜 배경에 수렴하는 문제점을 해결 할 수 있다.

나. 명암 대비를 이용한 컨트라스트 에너지

위에서 제시된 방향성분을 고려한 에지 에너지를 도입하여도 객체 윤곽선에 존재하는 기울기 성분과 유사한 배경 기울기 성분으로 인한 배경 잡음은 남을 수 있다. 이러한 배경 잡음에 강인하게 대처하기 위하여 객체의 윤곽선 정보를 이용하여 배경 잡음에 강인한 컨트라스트 에너지를 추가한다.

앞에서 보인 그림 3을 참고하여 컨트라스트 에너지를 계산하는 방법을 설명한다. 먼저  $s$  번째 스네이크 포인터에 대해  $s-1, s, s+1$  번째 스네이크 포인터를 잇는 스플라인 경계선을 구한다. 그림 3에서 보인 바와 같이  $b^{s,i}(j)$ 가 탐색윈도우 내의 픽셀  $w^s(i)$ 를 중심 좌표로 하는 서브블록윈도우 속에 속한  $j$  번째 픽셀 좌표를 나타낸다고 하자. 위에서 구한 스플라인 경계선이 평행이동 하여 이 서브블록윈도우의 중심점을 통과한다고 가정할 때 (그림 3에서 ‘스플라인 경계’로 표시) 이 가상 스플라인 경계선을 기준으로 서브블록윈도우 내의 각 픽셀이 스플라인 경계면의 Inner영역에 속하는지 또는 Outer영역에 속하는지를 나타내는 함수로  $f_{Inner}(b^{s,i}(j))$ 와  $f_{Outer}(b^{s,i}(j))$ 를 다음과 같이 정의한다.

$$f_{Inner}(b^{s,i}(j)) = \begin{cases} 1, & b^{s,i}(j) \in Inner \\ 0, & b^{s,i}(j) \notin Inner \end{cases} \quad (7)$$

$$f_{Outer}(b^{s,i}(j)) = \begin{cases} 1, & b^{s,i}(j) \in Outer \\ 0, & b^{s,i}(j) \notin Outer \end{cases} \quad (8)$$

이제 컨트라스트 에너지  $E_{contrast}^s$  는 식 (9)와 (10)을 이용하여 서브블록윈도우 내 Inner와 Outer영역의 평균 밝기  $\mu_{Inner}$ 와  $\mu_{Outer}$ 을 각각 구하여 식 (11)과 같이 각 영역 평균 밝기 차이의 절대값으로 계산한다. 식에서  $Y(b^{s,i}(j))$ 는 서브블록윈도우 내 픽셀의 밝기를 나타낸다.

$$\mu_{Inner} = \frac{\sum_{j=1}^M Y(b^{s,i}(j)) f_{Inner}(b^{s,i}(j))}{\sum_{j=1}^M f_{Inner}(b^{s,i}(j))} \quad (9)$$

$$\mu_{Outer} = \frac{\sum_{j=1}^M Y(b^{s,i}(j)) f_{Outer}(b^{s,i}(j))}{\sum_{j=1}^M f_{Outer}(b^{s,i}(j))} \quad (10)$$

$$E_{contrast}^s(u(i)) = - |\mu_{Inner} - \mu_{Outer}| \quad (11)$$

그림 5는 컨트라스트 에너지 생성 예이다. 그림 5의 (a)와 같이 사각형 영역의 중심부에 스네이크 포인터가 존재 할 때, (b)는 탐색윈도우 내의 어떤 서브블록윈도

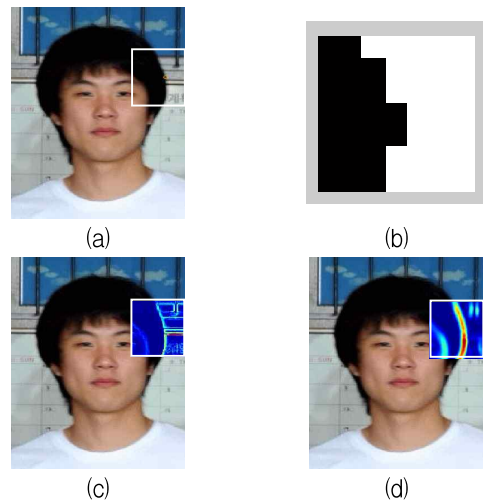


그림 5. 컨트라스트 에너지 생성 예 (a)스네이크 포인터, (b)서브블록윈도우, (c)기울기 맵, (d)컨트라스트 에너지 맵

Fig. 5. An example of contrast energy generation. (a) snake pointer, (b) a sub-block region, (c) gradient map, (d) contrast energy map.



우를 보이는 것으로 가상 스플라인 경계선을 기준으로 검은색이 Inner 영역, 흰색이 Outer 영역을 나타내고 있다. (c)는 밝기 값의 기울기 맵을 나타내고 있으며, (d)는 제안된 컨트라스트 에너지  $E_{contrast}^s$  맵을 나타낸다. (c)에 비해 (d)가 배경보다 객체 윤곽선을 더 잘 표현하고 있음을 알 수 있다.

## 2. 차영상의 윤곽선 정보를 이용하는 모션 벡터

객체가 급격하게 움직일 때, 그림 2의 (b)와 (c)에 보인 바와 같이 탐색 윈도우가 객체의 윤곽에 도달하지 못하면 윤곽선이 객체 내부로 잘못 수렴(즉, 수축)된다. 이러한 문제를 해결하기 위하여 본 논문에서는 이전 프레임과 현재 프레임의 차영상의 윤곽선 정보를 이용하는 모션 벡터를 생성하여 이전 프레임의 스네이크 포인터를 현재 프레임의 객체 윤곽 인근으로 사전에 이동시켜 탐색 윈도우가 객체의 윤곽에 도달하지 못하는 것을 방지하며 빠르게 객체 윤곽에 수렴하게 하는 방법을 제안한다.

제안하는 모션 벡터는 다음과 같이 구한다. 먼저 이전 프레임과 현재 프레임의 차영상  $D$ 를 구한다. 차영상  $D$ 에는 이전 프레임과 현재 프레임에서 밝기 차이가 큰 부분이 드러나게 되므로 인접한 두 프레임의 움직임이 잘 나타나게 된다. 다음으로 이전 프레임에서 얻은 각 스네이크 포인터에 대하여 탐색윈도우를 설정하고 탐색윈도우 내의 각 픽셀에 대하여 서브블록윈도우를 설정한 후 전술한 컨트라스트 에너지  $E_{contrast}^s$ 를 구하는 과정과 똑같은 방법으로 서브블록윈도우 내의 가상 스플라인 경계선을 구한다(그림 3과 III.1.나.절 참조). 마지막으로 서브블록윈도우 픽셀  $b^{s,i}(j)$  중 가상 스플라인 경계선 위에 존재하는 차영상  $D$ 의 밝기 값의 합을 탐색윈도우 내의 모든 픽셀에 대하여 계산한 후 가장 그 값이 큰 픽셀의 좌표를 식 (12)를 사용하여 구한다. 이때 스네이크 포인터  $s$ 로부터 멀리 떨어진 탐색윈도우 내  $i$ 번째 픽셀이 우선적으로 선택이 되도록 식 (13)에 보인 바와 같은 가중치  $W^s(i)$ 를 부여한다. 이것은 차영상  $D$ 에서 객체의 움직임에 따라 이전 프레임에 구한 윤곽선 대신 현재 프레임에서 변화된 윤곽선을 선택하기 위한 것이다. 여기서  $f_{boundary}(b^{s,i}(j))$ 는 서브블록윈도우 내 픽셀  $b^{s,i}(j)$ 가 가상 스플라인의 경계 위에 존재하는지 시험하는 함수이다.

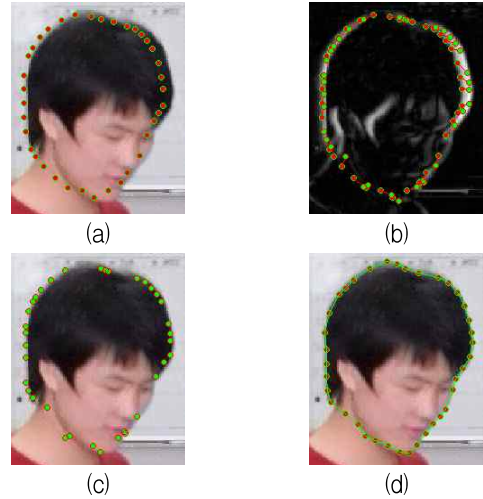


그림 6. 모션 벡터를 적용한 얼굴 추적 예 (a)이전 프레임의 스네이크 포인터, (b)차영상위의 스네이크 포인터들, (c)모션 벡터를 적용 후 스네이크 포인터, (d)윤곽 추적 결과

Fig. 6. An example of face tracking using motion vectors. (a)snake pointer of previous frame, (b)snake pointers on the difference image, (c)snake pointer after applying motion vectors, (d)result of tracking contour.

$$(x(\hat{i}), y(\hat{i})) = \arg \underset{i}{MAX} (W^s(i) \sum_j^M D(b^{s,i}(j)) f_{boundary}(b^{s,i}(j))) \quad (12)$$

$$W^s(i) = \sqrt{(x(s) - x(i))^2 + (y(s) - y(i))^2} \quad (13)$$

이전 프레임의 객체의 윤곽을 나타내는 스플라인 경계의 사용은 이전 프레임의 스네이크 포인터를 객체의 윤곽과 유사한 영역으로 이동함으로써 객체의 급격한 이동으로 인하여 포인터가 객체의 윤곽에 도달하지 못하고 수축되는 현상을 방지한다.

그림 6은 제안된 전체 스네이크 추적 알고리즘을 적용한 얼굴 추적의 예이다. (a)는 이전 프레임의 스네이크 포인터들을 나타낸다. (b)는 이전 프레임과 현재 프레임의 차영상 위에 기존 스네이크 포인터와 함께 식 (12)에 의해 계산된 픽셀 좌표로 이동된, 즉 모션 벡터를 적용한 스네이크 포인터를 함께 표시하였다. (c)에는 현재 프레임 영상위에 모션 벡터를 적용하여 이동된 스네이크 포인터들만을 나타내었다. (d)는 (c)의 결과에 제안된 에너지 함수를 사용한 스네이크 알고리즘을 적용하여 얼굴 윤곽을 추적한 결과이다. 이 예를 통해 제안된 모션 벡터 사용 방법이 스네이크 포인터를 현재 프레임에서 윤곽선이 존재하는 부근으로 스네이크 포인

터를 미리 이동시키므로 얼굴의 윤곽선을 정확하게 추적 할 수 있음을 알 수 있다.

#### IV. 실험 및 결과 분석

실험은 인텔 Q6600 2.4GHz CPU, 3GB 메모리를 장착한 PC에서 Matlab 환경 하에서 수행되었다. 실험 영상은 3가지이며 각각 640x424 해상도를 갖는다. 실험에서는 스네이크 알고리즘들 중에 기본이 되는 ACM 기법<sup>[6]</sup>과 기존의 연구 결과 중 GVF, HDC 기법보다 성능이 더 우수한 두 단계 합성 기울기(TCG) 맵 기법<sup>[11]</sup>, 그리고 본 논문에서 제안하는 기법을 비교 분석하였다.

그림 7, 8, 9는 각 기법을 적용한 얼굴 추적 비교 결과이며 편의상 원 영상에서 추적 대상인 얼굴 부분을

편집하여 보였다. 그림 7은 주변 배경에 강한 기울기 성분이 존재하는 고정 카메라 영상을 사용한 얼굴 추적 결과이다. ACM 기법은 밝기 값의 기울기 성분의 크기에 크게 의존하기 때문에 얼굴이 움직이면 (e), (f)와 같이 배경에 수렴 되어 추적이 실패한다. TCG기법은 이진 차별 마스크를 적용하여 배경을 효과적으로 제거해 배경에 수렴하지 않고 추적하지만 (e)와 (f) 같이 프레임 간 객체 움직임이 존재 하지 않을 때, 이진 차별 마스크의 오류로 인하여 (f)에서처럼 성능 저하를 보이게 된다. 제안하는 기법을 적용한 결과는 배경에 수렴되지 않고 정확하게 추적하는 것을 볼 수 있다.

그림 8은 얼굴이 회전하는 고정카메라 영상의 추적 결과로 ACM 기법은 (b)와 같이 배경에 수렴되며 또 얼굴의 급격한 이동으로 인하여 탐색 윈도우가 객체 윤

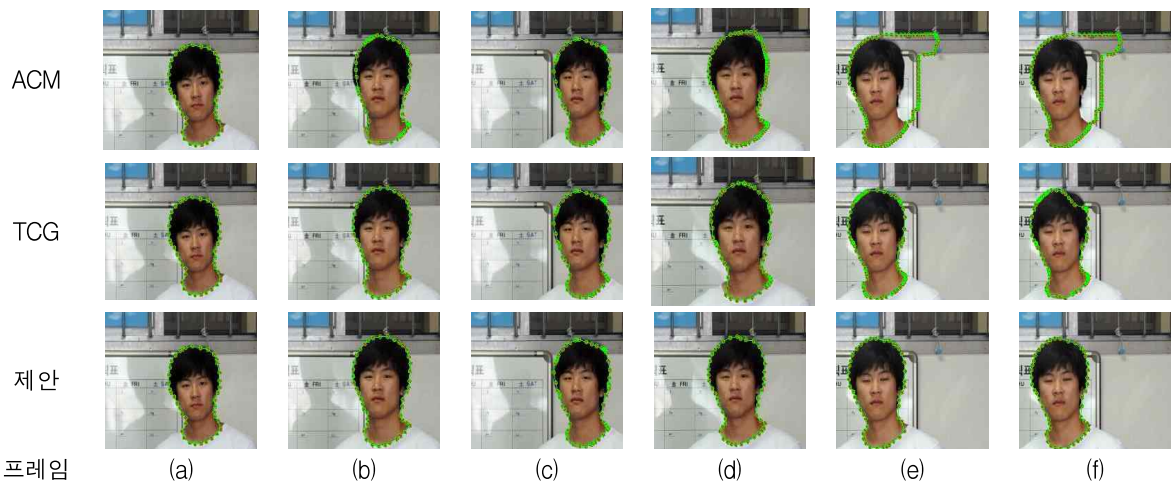


그림 7. 강한 배경 기울기 성분이 존재하는 영상의 추적 결과

Fig. 7. Tracking results of video that has background with a strong gradient component.

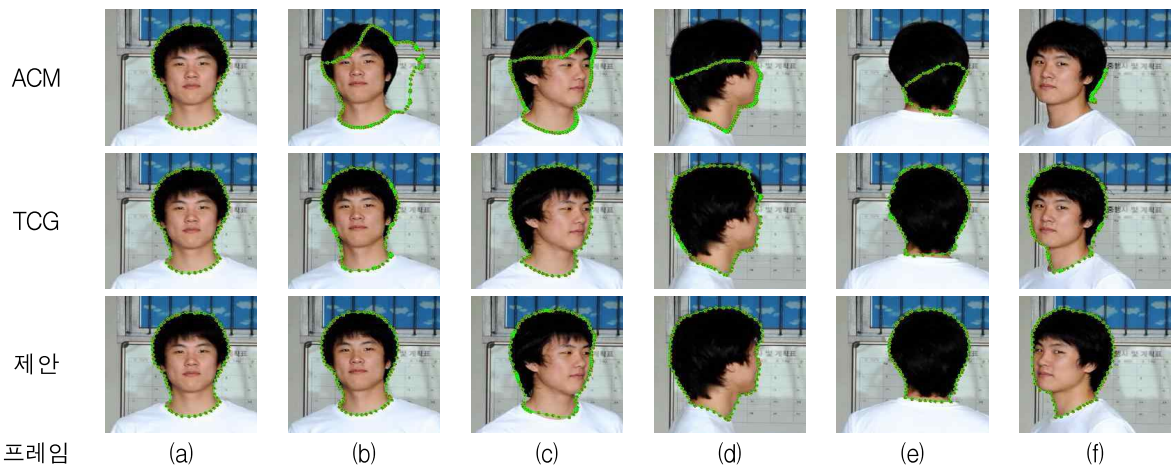


그림 8. 회전하는 얼굴 영상의 추적 결과

Fig. 8. Tracking results of face rotating video.

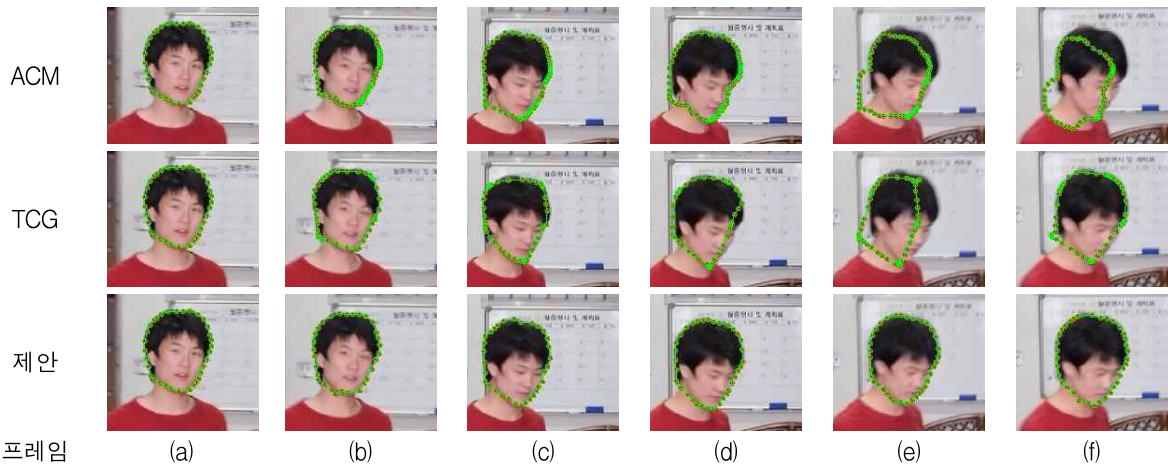


그림 9. 이동 카메라 영상의 추적 결과  
 Fig. 9. Tracking results of moving camera video.

곽에 도달하지 못해 (b)-(f)와 같이 얼굴 내부로 윤곽선이 수렴됨을 볼 수 있다. TCG 기법의 경우 (d)에서 성능 저하를 보이고 있다. 제안하는 기법은 얼굴이 회전하여도 얼굴의 윤곽을 정확하게 추적하고 있다.

마지막으로 그림 9는 얼굴이 정면에서 측면으로 움직이는 이동 카메라 영상의 추적 결과이다. 이동 카메라 영상은 고정 카메라 영상보다 움직임이 불규칙 하며 영상에 모션 블러가 생겨 기울기 성분들이 약하게 나타는 특징이 있다. ACM 기법의 경우 얼굴이 이동하면서 (e)와 같이 배경에 잘못 수렴되며, 얼굴에 블러 효과와 급격한 이동으로 인해 (e), (f)와 같이 객체 내부로 수렴되는 것을 확인 할 수 있다. TCG 기법을 이용한 추적에서는 이동 카메라 영상이기 때문에 이진 차별 마스크의 정확도가 떨어지게 되어 (e)와 같이 배경으로 잘못 수렴한다. 하지만, TCG 기법은 HDC 기법과 합성 기울기 맵을 결합하였기 때문에 (f)와 같이 오류를 정정할 수 있는 능력을 가지고 있다. 제안하는 기법을 사용한 추적에서는 이동 카메라 영상에서도 배경과 윤곽의 기울기 성분을 구분하여 배경 잡음에 견고하고 또 큰 움직임이 존재 할 때 모션 벡터를 사용하여 얼굴의 윤곽의 움직임을 정확하게 추적하는 것을 확인할 수 있다.

그림 10은 실험에 적용한 각 기법들의 정확한 성능 비교를 위하여, 그림 9에서 보인 이동 카메라 영상의 매 프레임마다 수동으로 얼굴의 윤곽선에 초기 스네이크 포인트를 지정한 후 스네이크 알고리즘을 적용해 찾은 정확한 스네이크 포인트들(Ground Truth)과 각 기법들을 적용하여 추적에서 얻어진 스네이크 포인트들의 200프레임 동안의 픽셀 오차를 나타내었다. 또, 표 1에

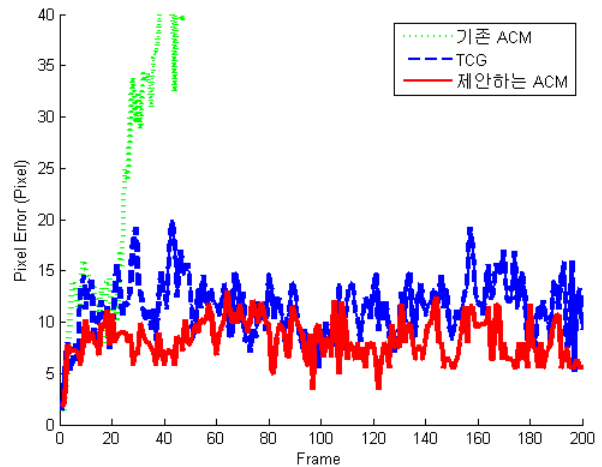


그림 10. 적용한 기법에 따른 픽셀 오차  
 Fig. 10. The pixel error of each algorithm.

는 200프레임 동안의 평균 픽셀 오차를 보였다. 이 결과 제안하는 기법이 기존 기법들 보다 프레임 당 픽셀 오차가 작으며, 픽셀 오차의 평균이 낮은 것을 확인할 수 있다. 따라서 기존 기법들보다 제안하는 기법이 더 안정적으로 추적하는 것을 보여준다.

표 2는 추적 객체의 픽셀 크기와 각 기법들의 프레임 당 수행 시간의 평균을 측정하여 비교한 결과이다. ACM 기법은 기울기 맵만 생성하므로 가장 적은 연산 시간을 필요로 한다. TCG 기법은 이진 차별 마스크와 다중 합성 기울기 맵, 단기 연산으로 인하여 ACM 기법보다 약 30배 정도의 연산 시간이 필요하다. 제안하는 기법은 윤곽선의 정보를 이용하여 기울기 성분들의 내적과 컨트라스트 에너지, 모션 벡터를 연산하므로 스네이크 포인트의 수에 따라 민감하게 반응하며, 객체의



표 1. 적용한 기법에 따른 평균 픽셀 오차  
Table 1. The average pixel error of each algorithm.

기법	ACM	TCG	제안
평균오차픽셀	추적실패	11.39	8.23

표 2. 기법에 따른 수행 시간 (초/프레임)  
Table 2. The average execution time of each algorithm.  
(sec/frame)

기법	객체 크기	108x160 pixels	150x230 pixels	95x121 pixels
	ACM	0.03	0.06	0.02
TCG		1.66	1.7	1.65
제안		2.08	2.72	1.86

윤곽이 큰 영상에서는 TCG 기법 보다 1.5배 정도 연산 비용이 증가되고 작은 영상에서는 1.13배 정도 연산 비용이 증가되지만 기존 방법들 보다 향상된 객체 추적이 가능하였다.

### V. 결 론

기존 스네이크 알고리즘을 이용한 객체 추적에서 배경에 강한 기울기 성분이 존재 하는 경우 윤곽선이 배경에 수렴하기 때문에 추적이 실패하게 된다. 또한 객체의 이동으로 인한 모션 블러 현상이 발생하여 배경과 객체 사이의 경계가 불확실해거나 급격한 객체의 이동으로 인하여 탐색 윈도우가 객체의 윤곽에 도달하지 못하여 윤곽선이 객체의 내부로 수축되는 현상을 겪게 된다. 이러한 기존 스네이크 알고리즘의 문제를 해결하기 위하여 본 논문에서는 스플라인 경계 위에 존재하는 기울기 성분과 탐색 윈도우 영역에 존재하는 기울기 성분을 내적함으로써 배경의 기울기와 객체 경계의 기울기 성분을 구분할 수 있는 방향 성분을 고려한 에지 에너지를 제안하고, 스플라인 경계를 기준으로 *Inner* 영역과 *Outer* 영역의 평균 밝기 차이를 이용하여 컨트라스트 에너지를 추가해 복잡한 배경에서 변형하는 객체의 윤곽을 정확하게 추적하였다. 또한 이전과 현재 프레임의 차영상을 기반으로 스플라인 경계에 존재하는 차영상 값이 가장 큰 픽셀로 스네이크 포인터를 이동하게 하는 모션 벡터를 사용하여 현재 프레임의 객체 윤곽선에 탐색 영역이 빠르고 정확하게 도달 할 수 있도록 개선하였다. 본 논문에서 제안하는 기법은 기존 기법보다 연산 비용이 약간 높지만 기존 기법들의 추적 한계를

극복하고 고정, 이동 카메라 영상 모두 안정적으로 정확하게 추적하였다.

### 참 고 문 헌

- [1] D. Comaniciu, V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," International Conference on Image Processing, pp. 70-73, 2000.
- [2] K. Nummiaro, E. Koller-Meier and L. Van Gool, "A Color-Based Particle Filter", First International Workshop on Generative-Model-Based Vision, pp. 53-60, June 2002.
- [3] P. Perez, C.hue, J.Vermaak and M. Gangnet., "Color-based probabilistic tracking," European Conference on Computer Vision, pp. 661-675, 2002.
- [4] E. Maggio, F. Smerladi, and A. Cavallaro., "Adaptive multi-feature tracking in a particle filtering framework," IEEE Trans. Circuits and Systems for Video Technology, 2007.
- [5] M. Kass, "Snake: Active Contour Model," International Journal of Computer Vision, vol. 1, pp. 321-331, 1988.
- [6] M. Pardas, E. Sayrol, "Motion Estimation based Tracking of Active Contours," Pattern Recognition Letters 22, pp.1447-1456, 2001.
- [7] Bing, X. Wei, Y. Charoensak, C., "Face Contour Tracking in Video using Active Contour Model," Image Processing, International Conference on Image Processing(ICIP), vol.2, pp.1024-1024, 2004.
- [8] Chenyang Xu, Prince, J.L., "Gradient Vector Flow: A New External Force for Snakes," Computer Vision and Pattern Recognition, Proceedings, IEEE Computer Society Conference on, pp. 66-71, 1997.
- [9] Leymarie, F. Levine, M.D., "Tracking Deformable Object in the Plane Using an Active Contour Model," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.15, no.6, pp.617-634, 1993.
- [10] J.H. Lee, H.G. Oh, H.Hong, "Active Contour Model for Object Tracking with Large Motion Displacement," Korea Computer Congress, vol.33, pp.464-469, 2006.
- [11] 김수경, 장유진, 홍헬렌, "두 단계 합성 기울기 맵을 이용한 활성 외곽선 모델 기반 자동 얼굴 추적," 한국정보과학회, vol.36, no.11, pp.901-911, 2009.
- [12] Blake, Andrew, "Active contours: the application

of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion”, Springer, 1999.

---

— 저 자 소 개 —

---



김진울(정회원)

1986년 서울대학교 전자공학과  
학사.

1988년 한국과학기술원 전기 및  
전자공학과 공학석사.

1993년 한국과학기술원 전기 및  
전자공학과 공학박사.

1994년~현재 수원대학교 전자공학과 부교수.  
<주관심분야: 신호처리시스템 설계, 영상 내 객체  
추적>



정재기(정회원)

2009년 수원대학교 전자공학과  
학사 졸업.

2011년 수원대학교 전자공학과  
석사 졸업.

현재 UDworks, Co. Ltd. 연구원.  
<주관심분야 : 객체 추적, 영상신  
호처리>