

오목한 윤곽을 갖는 객체에서 스네이크 기반의 윤곽선 검출 방법

김 신 형[†] · 장 종 환^{**}

요 약

본 논문에서는 스네이크 기반의 객체 윤곽선 검출 방법을 제안한다. 기존의 방법들은 스네이크 에너지 함수의 제약으로 오목한 윤곽을 갖는 복잡한 모양의 객체에 대해서는 윤곽선 검출의 정확도가 떨어지고, 고정된 스네이크 포인트 수를 이용하기 때문에 효율적으로 객체 윤곽을 표현하는데 문제점이 있다. 이러한 문제점을 해결하기 위해 스네이크 포인트의 추가 및 제거를 통해 객체의 윤곽을 검출하는 방법을 제안한다. 제안한 방법은 크게 두 단계로 구성된다. 먼저, 제안하는 스네이크 모델로 초기 관심객체의 윤곽을 검출하고, 두번째 단계로 부족한 포인트는 추가하고 잉여 포인트는 제거하여 객체의 윤곽선을 잘 검출할 수 있게 한다. 제안한 방법은 복잡한 윤곽을 갖는 객체에 대해 윤곽선 검출이 기존의 방법보다 좋은 결과를 보이는 것을 실험을 통해 확인하였다.

키워드 : 스네이크, 윤곽검출, 오목한 윤곽, 프레네 방정식

A Snake-Based Segmentation Algorithm for Object with Boundary Concavities

Shin-Hyoung Kim[†] · Jong-Whan Jang^{**}

ABSTRACT

In this paper we present a snake-based scheme for efficiently detecting contours of objects with boundary concavities. The proposed method is composed of two steps. First, the object's boundary is detected using the proposed snake model. Second, snake points are optimized by inserting new points and deleting unnecessary points to better describe the object's boundary. The proposed algorithm can successfully extract objects with boundary concavities. Experimental results have shown that our algorithm produces more accurate segmentation results than the conventional algorithm.

Key Words : Snake, Contour Detection, Boundary Concavities, Frenet Formulas

1. 서 론

최근 멀티미디어 서비스의 형태는 다양한 서비스를 요구하는 수요자들을 위해 내용에 기반한 대화형 멀티미디어 서비스(content-based interactivity)[1]로 발전하는 추세이다. 이러한 대화형 멀티미디어 서비스를 제공하기 위해서는 객체단위 부호화가 필수적이고, 객체 부호화를 하기 위해서는 먼저 객체를 성공적으로 분할해야 한다. 특히, 멀티미디어 표준안으로 제정된 MPEG-4에선 비디오 객체 평면(video object plane)이란 개념을 소개하고 내용기반 부호화 방식을 채택하였다[2]. 이렇듯 객체 분할은 대화형 멀티미디어 시스

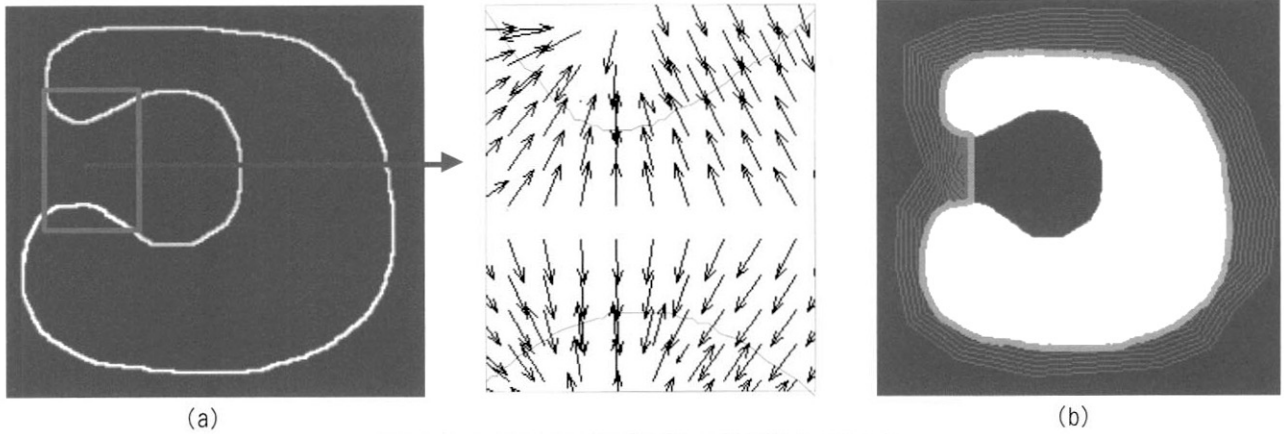
템 및 비디오 코딩 그리고 다양한 컴퓨터 비전분야의 중요한 전처리 작업이다.

일반적으로 널리 알려진 2차원 영상에서의 객체분할에 관한 연구는 화소들간의 유사성 측정에 의한 임계치 방법, 영역 성장, 영역 분할 및 병합 등[3-7] 있고 정의된 에너지 함수를 최소화하는 과정을 통해 객체의 윤곽선을 추출하는 스네이크(snake) 모델이 잘 알려져 있다[8]. 특히, 스네이크는 간단하면서도 효과적으로 객체의 윤곽을 검출하는 장점이 있어 객체 분할 및 추적이 많이 사용되고 있다. 그러나 초기의 방법은 객체의 윤곽이 복잡하고 오목하게 들어간 윤곽 부분에서는 에너지 부족 현상으로 인해 정확한 윤곽선 검출이 어렵다[9]. 또한 고정된 수의 스네이크 포인트를 이용하기 때문에 큰 곡률(curvature)을 갖는 부분에서 객체 윤곽을 정확히 표현하기에는 문제점이 있다. 이러한 문제점을 해결

[†] 준 회원: 배재대학교 정보통신공학과 박사과정

^{**} 종신회원: 배재대학교 정보통신대학원 원장

논문접수: 2006년 2월 23일, 심사완료: 2006년 7월 13일



(그림 1) 호리병 모양 윤곽에서의 GVF 에너지 집중현상

하기 위해 Xu[10]는 GVF(gradient vector flow) 스네이크를 제안하였다. 이 방법은 오목한 윤곽을 갖는 객체에 적용할 수 있는 모델로서 외부에너지로 단순히 에지정보를 이용하지 않고 에지의 공간 확산을 통해 얻은 GVF 에너지를 이용해 객체의 오목한 윤곽까지 검출할 수 있었다. 그러나, 이 방법은 GVF 에너지 맵을 구하기 때문에 많은 계산시간을 필요로 할 뿐만 아니라 호리병 모양(gourd shaped)의 윤곽을 갖는 객체에 대해서는 좁은 입구(neck of gourd)에서 GVF의 에너지의 집중현상으로 인해 안쪽 윤곽까지 수렴하지 못하는 문제점이 있다. 이러한 문제점들을 해결하고 객체의 윤곽을 잘 검출하기 위한 새로운 스네이크 모델과 스네이크 포인트의 추가 및 제거 과정을 통해 보다 정확한 객체 윤곽선 검출 방법을 본 논문에서 제안한다.

제안한 객체 윤곽선 검출 방법은 크게 두 단계로 이루어진다. 먼저, 제안하는 스네이크 모델을 이용해 초기 윤곽선을 검출한다. 스네이크 모델은 내부에너지에 미분기하학의 프레네 방정식(Frenet formulas)을 적용하여 스네이크 포인트에서의 종법선벡터(binormal vector)[11]를 이용해 오목한 윤곽까지 수렴하게 된다. 두번째 단계는 수렴된 스네이크 포인트들에서 곡률벡터(curvature vector)를 구해 스네이크 포인트의 추가 및 제거 과정을 반복 수행하여 복잡한 객체에서 객체의 윤곽을 더 정확하게 검출한다.

논문의 구성은 2장에서 스네이크의 기본 알고리즘에 대해 알아보고, 3장에서 객체 윤곽 검출을 위해 제안하는 스네이크 알고리즘과 스네이크 포인트 추가 및 제거에 대한 방법을 설명한다. 제안한 알고리즘의 성능분석은 4장에서 기술하고, 5장에서 결론을 맺는다.

2. 스네이크 알고리즘

스네이크는 처음 Kass[8]에 의해 소개되었고 사용자가 입력한 스네이크 포인트로부터 에너지함수를 반복적으로 최소화하는 방향으로 움직이면서 객체의 윤곽을 찾아내는 방법이다.

스네이크 알고리즘을 이산적으로 표현하면 다음과 같다. 스네이크 포인트를 $v_i = (x_i, y_i)$ 라 하면, $i = 0, \dots, M - 1$ 로 M

은 전체 스네이크 포인트의 개수이고, x_i, y_i 는 i 번째 스네이크 포인트의 x, y 좌표이다. 스네이크 포인트를 이동하게 하는 에너지 함수는 식 (1)과 같이 내부 에너지(internal energy) $E_{int}(v_i)$ 와 외부 에너지 (external energy) $E_{ext}(v_i)$ 의 합으로 나타낸다.

$$E_{snake}(v) = \sum_{i=0}^{M-1} (E_{int}(v_i) + E_{ext}(v_i)) \quad (1)$$

내부 에너지는 식 (2)와 같이 스네이크 포인트의 1차 미분과 2차 미분의 합으로 이루어진다. 내부에너지의 최소화 과정은 스네이크 포인트를 연결하는 곡선을 부드럽게 한다.

$$E_{int}(v_i) = \alpha \|v_i - v_{i-1}\|^2 + \beta \|v_{i-1} - 2v_i + v_{i+1}\|^2 \quad (2)$$

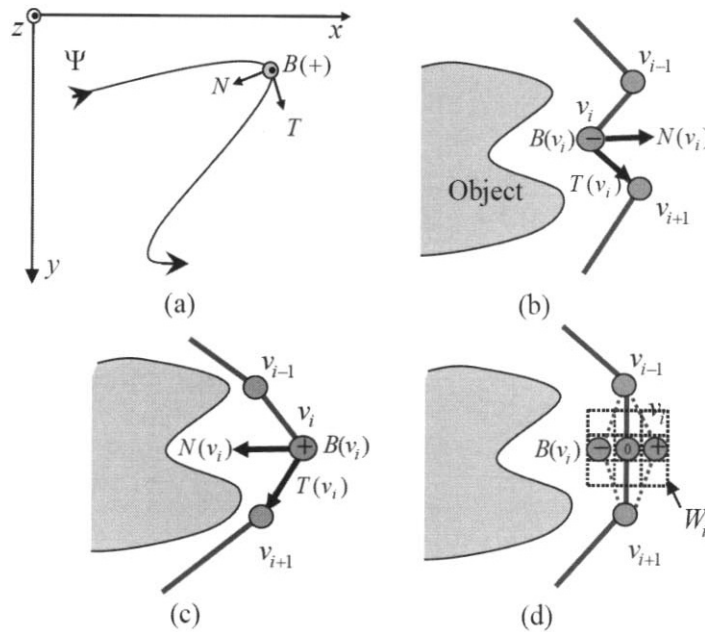
여기서 2차 미분항이 curvature 에너지를 나타내는데, 이 에너지는 현재 포인트와 이웃한 포인트가 직선을 만들 때 최소값 '0'을 갖는다. 오목한 윤곽이 존재하는 객체의 경우 안쪽 윤곽으로 스네이크 포인트를 이동시켜야하지만, 위와 같은 curvature 에너지의 제약으로 인해 더 이상 이동을 하지 못하는 문제점을 갖고 있다.

전통적인 외부 에너지는 식 (3)으로 나타내고 영상에서 경사도가 큰 곳에서 작은 값을 갖게 하여 스네이크 포인트가 관심객체(OOI: Object Of Interest)의 윤곽에 놓이도록 한다. 일반적으로 영상의 에지 정보를 사용한다.

$$E_{ext}(v_i) = -\gamma |\nabla f(v_i)|^2 \quad (3)$$

매개변수인 α, β 그리고 γ 의 가중치를 줌으로써 내부 및 외부 에너지의 중요도를 결정한다. 일반적으로 γ 가 다른 두 값보다 크게 설정함으로써 스네이크 포인트가 객체의 윤곽에 놓이게 한다.

오목한 윤곽에도 스네이크 포인트가 잘 수렴하도록 하기 위해 제안한 Xu의 GVF 스네이크는 외부에너지를 일반 에지 맵이 아닌 GVF 에너지 맵을 적용하여 오목한 윤곽에 적용할 수 있는 방법을 제안하였다. 그러나, (그림 1)의 (a)와 같이



(그림 2) 스네이크 포인트의 위치에 따른 종법선벡터의 방향

호리병 모양의 객체 윤곽에서는 좁은 입구에서 GVF 에너지가 집중되는 현상을 보인다. 따라서, (b)와 같이 스네이크 포인트가 안쪽의 윤곽까지 이동하지 못하고 호리병 모양 윤곽의 입구에서 수렴하는 문제점을 갖고 있다.

3. 제안하는 알고리즘

본 장에서는 기존의 스네이크 모델을 변경하여 새로운 에너지를 제안하고, 스네이크 포인트의 추가 및 제거 방법을 통해 객체의 윤곽을 더 효과적으로 검출하는 알고리즘을 제안한다.

3.1 제안하는 스네이크 에너지 함수

스네이크 에너지는 크게 내부에너지와 외부에너지로 구성이 되고, 내부에너지는 continuity 와curvature 에너지로 구성된다. 본 논문에서는 객체의 오목한 윤곽에 잘 적용될 수 있는 내부에너지를 제안한다.

3.1.1 내부에너지

내부에너지의 하나인 continuity에너지는 현재 스네이크 포인트 v_i 와 다음 스네이크 포인트 v_{i+1} 간의 거리가 전체 스네이크 포인트간의 평균거리에 의해 최소화 되면서 M 개의 스네이크 포인트들이 등 간격을 유지하도록 하는 에너지이다. 그러나 이러한 기존의 방법은 객체의 오목한 윤곽에서 문제점이 있다. 현재 포인트와 다음 포인트만고려할 경우, 오목한 윤곽에서는 스네이크 포인트 간의 거리가 다시 늘어나게 되기 때문에 등 간격을 유지 못하게 된다. 따라서, 전체 포인트의 평균거리를 사용하지 않고 현재 스네이크 포인트의 이웃한 스네이크 포인트들간의 거리만을 이용하여 다

음과 같은 에너지를 제안한다.

$$E_{con}(v_i) = \frac{\|v_{i-1} - v_i\| - \|v_i - v_{i+1}\|}{con_{max}} \quad (4)$$

여기서 con_{max} 는 정규화를 위해 현재 스네이크 포인트 v_i 를 중심으로 v_i 를 포함한 검색 윈도우(W_i)내 포인트들 중 최댓값을 나타낸다.

두번째 curvature 에너지는 현재 스네이크 포인트와 이웃한 스네이크 포인트가 이루는 벡터의 합의 크기로 나타낼 수 있고, 그 값이 작은 쪽으로 스네이크 포인트를 이동시킨다.

일반적으로 초기 스네이크 포인트는 객체의 외부에 설정하고 내부로 수렴 되는데, 객체의 오목한 윤곽에서는 스네이크 포인트가 윤곽에 도달하기 전에 직선을 이루는 구간에서 curvature 에너지가 최소값 '0'이 되어 오목한 윤곽까지 수렴되지 못하는 문제점이 있다. 이러한 문제를 해결하기 위해 스네이크 포인트에서의 종법선벡터를 분석하여 포인트의 이동을 제어하는 방법을 제안한다.

(그림 2)의 (a)는 $\Psi : I \rightarrow \mathbb{R}^3$ 이 단위 스피드 곡선이라 할 때, 변이 공간상에서의 곡선 위의 접선벡터(tangent vector) T 와 법선벡터(normal vector) N , 종법선벡터 B 를 표현하였다. 각각은 프레네 방정식을 이용해 식 (5)로 나타낸다. 단, $T \cdot T = 1, T \cdot T' = 0$ 이다.

$$T = \Psi', \quad N = \frac{T'}{\|T'\|}, \quad B = T \times N \quad (5)$$

여기서 B 의 부호에 대해 생각해 볼 수 있다. B 가 위쪽을 향하면 부호가 양이고 아래를 향하면 음이다. 종법선벡터의 부호를 2차원에서 확인하기 위해 식 (6)과 같이 접선벡터와

법선벡터의 외적을 취한다.

$$B(v_i) = T(v_i) \times N(v_i) = \begin{vmatrix} x_i^T & y_i^T \\ x_i^N & y_i^N \end{vmatrix} = (x_i^T y_i^N - y_i^T x_i^N) \vec{e}_z \quad (6)$$

여기서 (x_i^T, y_i^T) 와 (x_i^N, y_i^N) 는 현재 포인트에서의 접선벡터 $T(v_i)$ 와 법선벡터 $N(v_i)$ 의 x 와 y 좌표를 나타내고, \vec{e}_z 는 z 성분의 단위 벡터를 나타낸다. 여기서 $T(v_i)$ 와 $N(v_i)$ 을 이산적으로 표현하면 다음과 같이 근사화 할 수 있다.

$$T(v_i) \approx v_{i+1} - v_i \quad (7)$$

$$N(v_i) \approx v_{i-1} - 2v_i + v_{i+1} \quad (8)$$

(그림 2)의 (b)와 (c)는 연속된 3개의 스네이크 포인트의 위치에 따른 포인트의 움직임을 설명한 것이다.

먼저, 현재 스네이크 포인트 v_i 를 중심으로 이웃한 스네이크 포인트들이 (그림 2) (b)와 같을 때, v_i 에서의 $T(v_i)$, $N(v_i)$ 그리고 $B(v_i)$ 는 그림과 같이 표현된다. v_i 가 객체의 윤곽을 향해 움직이도록 하기 위해 W_i 내 v_i 의 이웃한 검색 포인트에서 $\|N(v_i)\|$ 의 값이 큰 위치로 이동 하면 객체의 윤곽 쪽으로 이동하게 된다. 이동을 결정하기 위해 $B(v_i)$ 의 방향을 이용하였다. $B(v_i)$ 가 아래쪽을 향할 때 $\|N(v_i)\|$ 의 값이 큰 검색 포인트로 현재 포인트를 이동 시킨다.

(그림 2)의 (c)와 같은 경우는 v_i 가 객체의 윤곽을 향해 움직이려면 $\|N(v_i)\|$ 의 값이 위와는 반대로 작은 값을 갖는 포인트로 이동해야 한다. 이와 같은 경우에 $B(v_i)$ 의 방향이 위쪽으로 향하고 있으므로 이를 통해 스네이크 포인트를 $\|N(v_i)\|$ 가 작은 값을 갖는 위치로 이동 시키면 된다.

위와 같이 스네이크 포인트를 이동시키기 위해선 curvature 에너지의 부호만 경우에 따라 변경시켜줌으로 쉽게 구현할 수 있다.

앞에서는 스네이크 포인트가 객체 외부에 있을 경우에 대해 설명하였고, 포인트 추가로 인해 포인트가 객체 내부에 존재할 경우에는 위의 조건을 반대로 적용하면 된다. 이렇게 각 경우를 B 의 방향을 이용해 구분할 수 있기 때문에 B 를 구하여 v_i 에서의 이동방향을 결정할 수 있으며, 객체의 오목한 윤곽을 향해 스네이크 포인트를 이동 시킬 수 있다.

정규화된 curvature 에너지는 식 (9)와 같고, 제안하는 curvature 향의 전체 에너지 $E_{cur}(v_i)$ 는 식 (10)과 같이 정의하였다.

$$E_c(v_i) = \| T(v_i)/\|T(v_i)\| - T(v_{i-1})/\|T(v_{i-1})\| \| \quad (9)$$

$$E_{cur}(v_i) = \lambda E_c(v_i) \quad (10)$$

여기서 매개변수 λ 는 식 (6)의 $B(v_i)$ 가 위쪽 방향일 경우 '+1' 로, 아래쪽 방향일 경우에는 '-1' 로 설정된다.

(그림 2)의 (d)를 보면 현재 스네이크 포인트 v_i 에서의

$\|N(v_i)\|$ 의 값은 이웃한 포인트 v_{i-1} 와 v_{i+1} 와 직선을 이루기 때문에 '0'이다. 그러므로 $B(v_i)$ 는 제로벡터(zero vector)가 된다. 이런 경우도 마찬가지로 검색 윈도우 W_i 내에 위치하는 검색 포인트들에서의 $B(v_i)$ 를 조사하면 v_i 의 왼쪽에 위치하는 포인트의 경우 $B(v_i)$ 의 부호가 음이 되고 오른쪽 포인트는 양수가 되기 때문에 식 (10)에 의해 값이 작은 왼쪽 포인트로 이동시킬 수 있다.

3.1.2 외부에너지

외부에너지는 표준편차 σ 를 갖는 2차원 가우시안 함수를 영상과 컨벌루션하여 식 (11)과 같이 정의 하였다. e_{max} 는 W_i 내 검색 포인트들 중 최대값을 나타낸다.

$$e_{max} = \frac{-|V[G_\sigma(v_i) * f(v_i)]|}{e_{max}} \quad (11)$$

제안하는 전체 에너지함수는 식 (12)와 같이 정의한다.

$$E_{snake}(v) = \sum_{i=0}^{M-1} (\alpha E_{con}(v_i) + \beta E_{cur}(v_i) + \gamma E_{ext}(v_i)) \quad (12)$$

여기서 매개변수 $\alpha = 1.0$, $\beta = 1.0$ 그리고 $\gamma = 1.2$ 의 가중치를 사용하였다.

3.2 스네이크 포인트 추가 및 제거

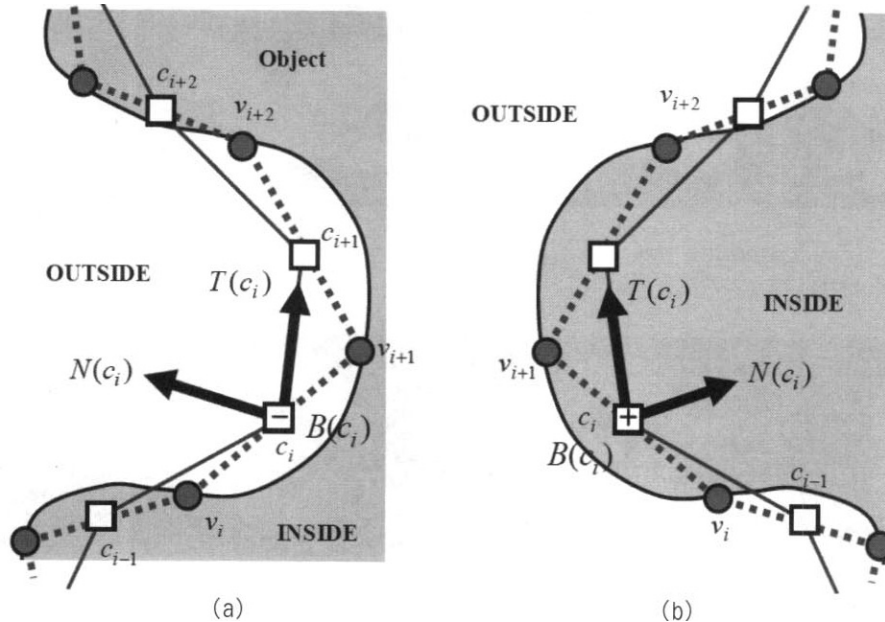
먼저 위에서 제안한 스네이크 모델을 이용해 1차 수렴을 수행한 후 스네이크 포인트가 객체의 윤곽에 놓여있다면, 그 윤곽을 잘 묘사하기 위해 부족한 스네이크 포인트를 추가해주고, 잉여 스네이크 포인트는 제거해 주는 과정을 통해 객체의 윤곽을 더 정확히 표현해 줄 필요가 있다.

3.2.1 스네이크 포인트 추가 및 제거

새로운 스네이크 포인트를 v_i 와 v_{i+1} 포인트 사이에 추가하는 결정은 식 (13)과 같이 $\|N(v_i)\|$ 와 $\|T(v_i)\|$ 을 이용한다. 만약 $\|N(v_i)\|$ 가 일정 임계값(th_N) 이상이고 $\|T(v_i)\|$ 가 임계값(th_T) 이상일 경우 새로운 스네이크 포인트를 추가한다.

$$\|N(v_i)\| \geq th_N \text{ AND } \|T(v_i)\| \geq th_T \quad (13)$$

이는 $\|N(v_i)\|$ 가 큰 값을 갖을 경우 현재 스네이크 포인트 v_i 와 이웃한 스네이크 포인트들 간의 곡률이 크기 때문에 윤곽을 더 정확히 검출하기 위해선 새로운 스네이크 포인트의 추가가 필요한 경우이다. 그리고 $\|T(v_i)\|$ 는 스네이크 포인트 간의 거리가 충분히 좁을 경우엔 포인트 추가가 불필요하기 때문에 스네이크 포인트 추가의 결정 조건으로 포함하였다. 여기서 새롭게 추가되는 스네이크 포인트는 $c_i = (v_i + v_{i+1})/2$ 로 정의 한다.



(그림 3) 객체 윤곽의 특징에 따른 스네이크 포인트 추가 및 제거 과정

반대로 $\|N(v_i)\|$ 가 임계값보다 작을 경우에는 현재포인트 v_i 가 이웃한 포인트가 이루는 선분이 직선에 가깝기 때문에 이 경우 v_i 를 제거한다. 이때, $\|T(v_i)\|$ 의 조건은 포함되지 않는다.

3.2.2 추가된 스네이크 포인트 위치에 따른 스네이크 모델 적용방법

(그림 3)은 추가된 스네이크 포인트가 객체 내부 혹은 외부에 존재할 경우에 객체의 윤곽으로 어떻게 이동하게 하는지 조사하고 있다. 스네이크 포인트가 추가 되었을 경우 그 포인트가 객체 내부에 있는지 혹은 외부에 있는지를 판단하기는 사전 정보가 없는 이상 쉬운 일이 아니다. 스네이크 포인트들만으로 객체 내부 또는 외부에 추가 되었다는 것을 판단해야 한다. 그 문제를 해결하는 방법으로 스네이크 포인트가 객체 내부나 외부에 존재할 때 그 포인트가 갖는 중법선벡터의 방향은 위치에 따라 반대로 나타난다. 특성을 이용하였다. 먼저, 추가된 스네이크 포인트들만을 이용하여 $B(c_i)$ 의 부호를 확인한다. 만약, $B(c_i)$ 가 음수이면, (그림 3)의 (a)와 같이 추가된 스네이크 포인트 c_i 는 객체 외부에 있다고 판단할 수 있다. 반대로 $B(c_i)$ 의 부호가 양수이면, (b)와 같이 객체 내부에 존재한다고 판단할 수 있다. 이 방법을 이용하여 추가된 스네이크 포인트의 움직임을 결정할 수 있다. 만약 추가된 스네이크 포인트가 객체 외부에 있다면 3.1절에서 설명한 제안한 방법을 똑같이 적용하면 추가된 포인트를 객체의 윤곽으로 이동시킬 수 있다. 반면에, 추가된 스네이크 포인트가 객체 내부에 존재하는 것으로 판단이 된다면 (즉, $B(c_i)$ 가 양수를 갖음) 3.1절에 소개된 방법에서 s_i 의 부호를 반대로 적용하여 수행하면 된다. 스네이크 수렴 과정에서는 모든 스네이크 포인트를 고려한다.

<표 1> (그림 5)의 RSD 및 속도 측정결과

	(a) Greedy 스네이크	(b) GVF스네이크	(c) 제안한 알고리즘
RSD	(5839/15143) 0.385	(3880/15143) 0.256	(128/15143) 0.008
속도(s)	0.073s	16.214s	0.109s

4. 실험 결과

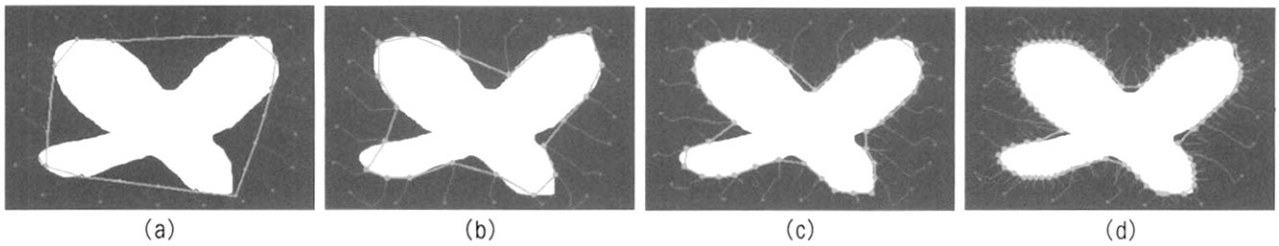
본논문에서 제안한 객체 추출 방법은 펜티엄-IV 3.0Ghz, 메모리 1Gbyte 환경에서 Visual C++6.0을 이용하여 구현 하였다. 실험은 이진 영상과 실 영상에 대해 객체의 모양이 복잡한 실험 영상을 사용하였다.

제안한 방법과 기존의 방법을 성능평가를 위해 Relative Shape Distortion (RSD)를 이용하여 예측된 윤곽선의 정확성을 정량화 하여 비교하였다. RSD는 식 (14)와 같이 나타낸다.

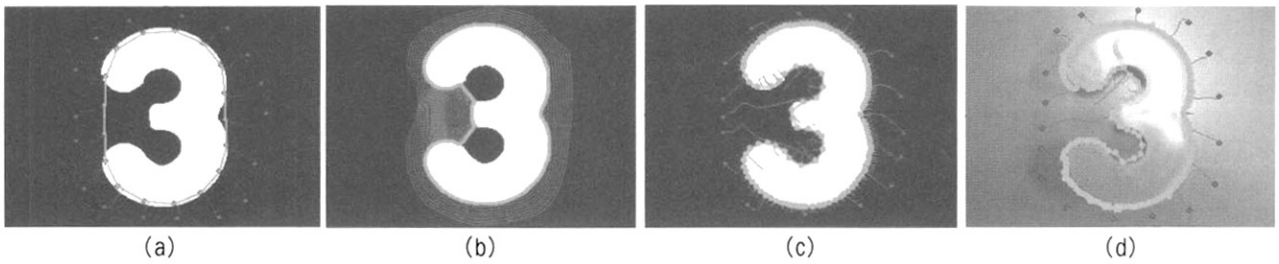
$$RSD(R) = \left(\frac{\sum_{(x,y) \in f} R_{ori}(x,y) \oplus R_{est}(x,y)}{\sum_{(x,y) \in f} R_{ori}(x,y)} \right) \quad (14)$$

여기서 R_{est} 는 예측된 객체 영역이 되고, R_{ori} 는 실제 객체의 영역이 된다. \oplus 는 이진 XOR 연산을 나타내고 실험결과를 <표 1>과 <표 2>에 나타내었다.

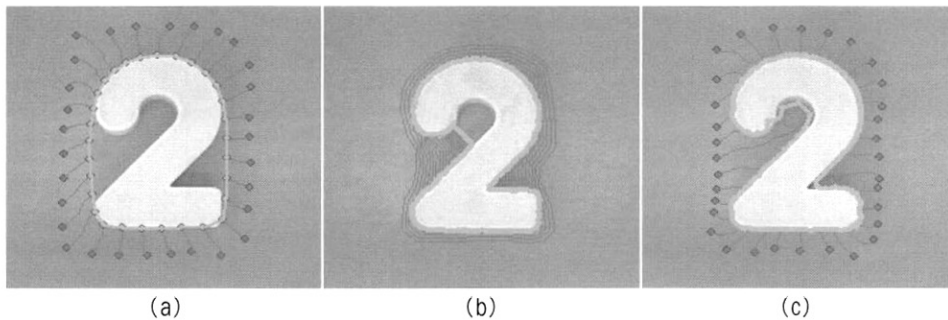
(그림 4)의 (a)는 기존 방법인 greedy 스네이크[3]를 이용해 객체의 윤곽을 검출한 실험결과로, 오목한 윤곽부분에서 에너지 부족 현상으로 인해 curvature 에너지가 '0'이 되는 지점에 수렴되기 때문에 오목한 윤곽에서는 포인트들이 직선을 이루는 결과를 보이고 오목한 윤곽으로 들어가지 못하는 것을 볼 수 있다. (그림 4)의 (b)-(d)는 제안한 방법의 실험 결과를 보여 주고 있다. (b)는 1차 수렴을 한 결과로 스네이크 포인트가 오목한 윤곽까지 찾아 들어가는 것을 볼 수



(그림 4) 기존 방법과 제안한 방법의 반복 수행결과 : (a) Greedy 스네이크, (b)-(d) 제안한 방법 (각각, 1차 수렴, 1차 포인트 추가 후 수렴, 2차 포인트 추가 후 수렴)



(그림 5) 숫자 3에 대한 기존 방법과 제안한 방법의 수행 결과 : (a) Greedy 스네이크, (b) GVF 스네이크, (c)-(d) 제안한 방법



(그림 6) 숫자 2에 대한 기존 방법과 제안한 방법의 수행 결과 : (a) greedy 스네이크, (b) GVF 스네이크, (c) 제안한 방법

<표 2> (그림 8)의 RSD 측정결과

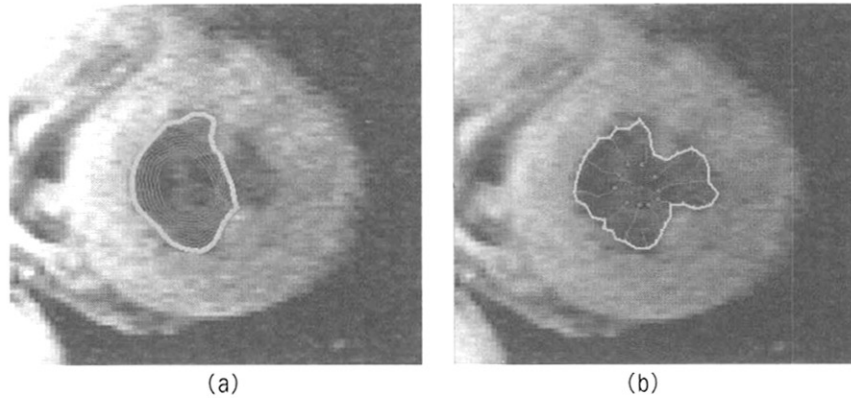
	초기 스네이크 포인트 수	반복 횟수 (추가 및 제거)	최종 스네이크 포인트 수	RSD
(a)	7	4	55	(459/25907) 0.017
(b)	14	3	58	(473/25907) 0.018
(c)	28	2	56	(311/25907) 0.012
(d)	56	1	43	(288/25907) 0.011

있다. 그러나 포인트 수가 부족해서 객체의 윤곽을 정확히 표현하지 못하였다. (c)는 포인트 추가 후 2차 수렴을 한 결과로 (b)보다는 객체의 윤곽을 잘 찾는 것을 볼 수 있고, 다시 추가 포인트를 추가하여 3차 수렴을 한 결과는 (c)에 보였다. 포인트가 추가되면서 객체의 윤곽을 더 잘 표현하는 것을 확인할 수 있었다.

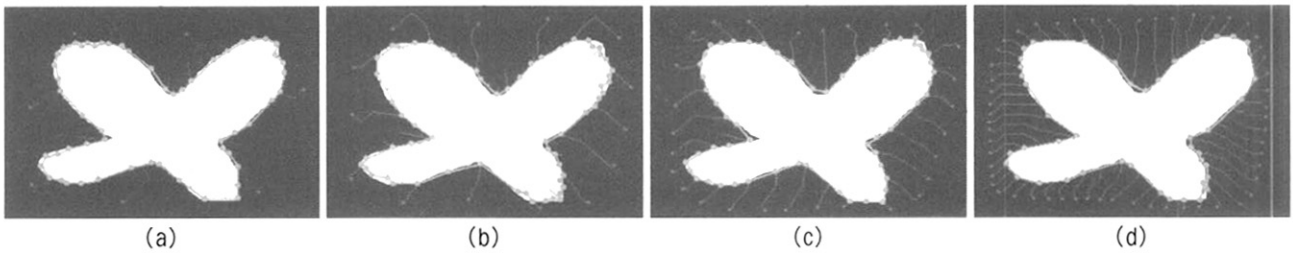
(그림 5)는 호리병 모양의 윤곽을 갖는 숫자 3에 대 실험을 수행한 결과이다. (그림 5)의 (a)는 greedy 스네이크를 수행한 결과로, 스네이크 포인트가 오목한 윤곽까지 수렴하지

못하고 직선을 이루는 것을 볼 수 있다. (b)는 GVF 스네이크를 적용한 결과로 호리병 모양의 윤곽까지 수렴되지 않는 것을 볼 수 있다. 반면에 제안한 방법은 네번의 수렴과정을 통해 (c)와 같이 객체의 윤곽을 잘 찾았음을 실험을 통해 확인 하였고, (d)와 같이 실 영상에서도 잘 동작하였다. 이 결과에 대한 RSD 측정 결과 및 알고리즘 수행시간을 <표 1>과 같이 나타냈다. Greedy 스네이크 방법과 제안한 방법은 그 수행시간이 1초 미만의 작은 값인 반면, GVF는 GVF 맵을 생성할 때 많은 계산시간을 요구하기 때문에 15초 이상의 수행시간이 걸리는 것을 확인할 수 있다. 숫자 2의 실 영상에 대한 실험결과를 (그림 6)에 보였다. Greedy 스네이크나 GVF 스네이크는 숫자 2의 오목한 윤곽을 잘 찾지 못했고, 특히 GVF 스네이크는 호리병 모양의 윤곽에서 더 이상 윤곽으로 이동하지 못하는 것을 볼 수 있다. 그러나 제안한 방법은 세번 반복과정을 통해 (c)와 같은 결과를 얻을 수 있었다. 호리병모양의 윤곽에도 스네이크 포인트가 이동되었고 반복 과정을 통해 윤곽을 비교적 잘 찾는 것을 확인할 수 있었다.

(그림 7)은 인간의 좌심실 심장 영상에 대해 실험을 수행한 결과를 보인 것이다. (그림 7)의 (a)는 GVF 스네이크, (b)는 제안한 방법에 대한 결과를 나타낸다. GVF 스네이크의



(그림 7) 인간의 심장 영상에서의 실험결과 : (a) GVF 스네이크, (b) 제안한 방법($\sigma = 2.5$)



(그림 8) 초기 스네이크 포인트 수와 추가/제거 반복 횟수와의 관계

경우, 초기 스네이크 포인트를 중앙의 돌출한 돌기(protrusion)를 포함되도록 설정을 하게 되면 논문[10]에서와 같이 윤곽을 잘 찾는 것을 볼 수 있다. 그러나 돌기를 포함하지 않고 (그림 7)과 같이 왼쪽 영역에 초기 스네이크 포인트를 설정하면 에너지 집중현상이 일어나는 돌기의 좁은 입구에서 더 이상 오른쪽 영역으로 이동하지 못하는 결과를 보였다. 반면, 제안한 방법은 전체적으로 영역을 잘 검출할 수 있었다.

(그림 8)은 초기 스네이크 포인트의 수와 포인트 추가/제거 반복 횟수에 대한 관계를 실험을 통해 보인 것이다. 적은 수의 초기 스네이크 포인트가 초기화 되었을 경우엔 몇 번에 스네이크 포인트 추가 및 제거과정을 반복함으로써 객체의 윤곽을 검출할 수 있고, 충분히 많은 초기 스네이크 포인트의 경우 보다 적은 반복횟수로 객체의 윤곽을 검출할 수 있었다. 이에 대한 비교는 <표 2>에 보였다. (그림 8)의 (a)는 7개의 초기 스네이크 포인트로 시작하여 네번의 반복을 통해 객체의 윤곽을 검출하였고, (d)의 경우 56개의 초기 스네이크 포인트로 한번 반복을 통해 객체의 윤곽을 검출하였다. 전체적으로 최종 스네이크 포인트의 수와 RSD 값은 비슷한 결과를 보였다. 하지만, 초기에 많은 스네이크 포인트를 이용할 경우엔 계산시간이 다소 많이 소요되는 문제점이 있기 때문에 적당한 수의 초기 포인트를 설정하고 추가 및 제거 과정을 통해 객체의 윤곽을 검출할 수 있다.

5. 결론 및 향후 연구

본 논문은 스네이크 기반의 객체 윤곽선 검출방법을 제안하였다. 제안한 방법은 크게 두 단계로, 첫번째 단계에서 종

법선벡터를 이용한 스네이크 모델을 적용하여 초기 객체 윤곽선을 검출하고, 두번째 단계는 스네이크 포인트의 곡률벡터를 이용하여 포인트 추가 및 제거를 수행하였다. 이를 통해 오목한 윤곽 또는 호리병 모양의 윤곽을 갖는 객체에서 에너지 부족현상으로 인해 객체의 윤곽을 잘 검출하지 못했던 기존의 문제점과 고정된 스네이크 포인트 수를 사용할 때 객체 윤곽선 표현의 어려움을 해결할 수 있었다. 제안한 방법과 기존 방법의 성능비교를 위해 이진 영상 및 실 영상에 대한 실험 결과를 보였고, 그 결과 기존 알고리즘에 비해 제안한 방법이 객체의 윤곽을 잘 검출 하는 것을 확인할 수 있었다. 제안한 방법은 단순한 배경의 영상에서 복잡한 윤곽을 갖는 객체를 대상으로 실험을 수행하였다. 영상 내 잡음의 영향을 줄이기 위해 외부에너지에 가우시안 함수를 적용하여 잡음의 영향을 줄일 수 있었지만, 향후 복잡한 배경(cluttered background)이나 강한 에지를 갖는 영상에서의 객체 윤곽 검출방법이나, 연구를 확장하여 객체 추적 등의 연구를 진행하고자 한다.

참 고 문 헌

[1] M. Bais, J. Cosmas, C. Dosch, A. Engelsberg, A. Erk, P. S. Hansen, P. Healey, G.K. Klungsoeyr, R. Mies, J.R. Ohm, Y. Paker, A. Pearmain, L. Pedersen, A. Sandvaned, R. Schafer, P. Schoonjans, and P. Stammnitz, "Customized Television : Standards Compliant Advanced Digital Television," IEEE, Trans. Broad., Vol.48, No.2, pp.151-158, June, 2002.

[2] ISO/IEC JTC/SC29/WG11/W4350: "Information Technology - Coding of Audio-Visual Objects Part2: visual" ISO/IEC 14496-2, July, 2001.

[3] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison Wesley Publishing Company, 1992.

[4] R. M. Haralick and L. G. Shapiro, "SURVEY: Image Segmentation Techniques," Computer Vision Graphics and Image Processing, Vol.29, 1985.

[5] R. Adams and L. Bischof, "Seeded Region Growing," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.16, No.6, pp.641-647, June, 1994.

[6] G. T. Herman and B. M. Carvalho, "Multiseeded Segmentation Using Fuzzy Connectedness," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.23, No.5, pp.460-474, May, 2001.

[7] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.13, No.6, pp.583-598, June, 1991.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snake: Active Contour Models," Int'l J. Computer Vision, Vol.1, No.4, pp.321-331, 1987.

[9] D. J. Williams and M. Shah, "A Fast Algorithm for Active Contours And Curvature Estimation," Computer Vision, Graphics, and Image Processing, Vol.55, pp.14-26, 1992.

[10] C. Xu and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow," IEEE Trans. Image Processing, Vol.7, No.3, pp.359-369, March, 1998.

[11] S.H. Kim and J.W. Jang, "Object Contour Tracking Using Snakes in Stereo Image Sequences," ICIP 2005, Vol.2, pp.414-417, pp.11-14, Sept., 2005.



김 신 형

e-mail : zeros@pcu.ac.kr
 2001년 배재대학교 정보통신공학과 (공학사)
 2003년 배재대학교 정보통신공학과 (공학석사)
 2003년~현재 : 배재대학교

정보통신공학과 박사과정

관심분야 : 영상처리, 컴퓨터비전, 멀티미디어



장 종 환

e-mail : jangjw@pcu.ac.kr
 1979년 한양대학교 전자통신공학과 (공학사)
 1986년 North Carolina 주립대학교 전기 및 컴퓨터공학과(공학석사)
 1990년 North Carolina 주립대학교 전기

및 컴퓨터공학과(공학박사)

1990년~현재 : 배재대학교 정보통신공학과 교수

2006년~현재 : 중소기업청 지정 배재대학교 창업보육센터 센터장

2003년~현재 : 배재대학교 정보통신대학원 원장

관심분야 : 영상처리, 멀티미디어 검색, 비디오편집, 컴퓨터 비전