

복수 객체의 윤곽 검출 방법에 대한 능동윤곽모델

장 종 환[†]

요 약

객체 윤곽을 추출하는 대부분의 기존 방법들은 단일객체의 윤곽검출에 대해 연구하였다. 그러나, 실 세계에서는 복수객체가 일반적이다. 본 논문에서 제안한 복수객체 윤곽추출 알고리즘은 2 단계로 구성된다. 첫 번째 단계는 외적 및 내적을 사용하여 초기에 복수객체를 포함한 윤곽을 고속으로 분리하고 연결하여 각각이 윤곽이 단일 개체만을 포함하는 방법을 제안한다. 두 번째 단계는 각각의 윤곽에 포함된 단일 객체의 윤곽을 추출하는 개선된 능동윤곽모델 알고리즘을 설명한다. 여러 실험영상에 대한 실험결과는 다른 방법과 비교하여 속도가 빠르며 정확하게 윤곽을 추출한다.

키워드 : 스네이크, 능동윤곽모델, 복수객체, 윤곽검출, 매우 불규칙한 윤곽, 스네이크 포인트의 분리 및 연결

Active Contour Model for Boundary Detection of Multiple Objects

Jong Whan Jang[†]

ABSTRACT

Most of previous algorithms of object boundary extraction have been studied for extracting the boundary of single object. However, multiple objects are much common in the real image. The proposed algorithm of extracting the boundary of each of multiple objects has two steps. In the first step, we propose the fast method using the outer and inner products; the initial contour including multiple objects is split and connected and each of new contours includes only one object. In the second step, an improved active contour model is studied to extract the boundary of each object included each of contours. Experimental results with various test images have shown that our algorithm produces much better results than the previous algorithms

Key words : Snake, Active Contour Model, Multiple Objects, Boundary Extraction, Highly Irregular Boundary, Splitting and Connecting of Snake Points

1. 서 론

객체윤곽 추출은 내용기반 검출시스템 및 대화형 멀티미디어시스템에서 매우 중요하다[1, 2]. 이러한 시스템을 사용하여 서비스를 성공적으로 제공하기 위해서는 영상질의를 하는데 기본 정보로 객체 모양을 사용한다. 실 영상에서는 단일객체 보다는 복수 객체가 일반적이고 복수객체 윤곽을 효율적으로 추출하면 활용되는 분야가 다양화 될 것으로 기대된다. 복수 객체의 윤곽추출의 일반적인 방법은 먼저 복수객체를 단일객체로 분리하고, 그 다음 단일 객체의 모양 즉 윤곽을 추출하여야 한다.

복수객체의 윤곽추출에는 여러 방법이 제안되었다. 예를 들면, 영역분할이나, Watershed 등의 알고리즘이 제안되었

지만, 본 논문에서는 복수객체의 윤곽추출을 위한 능동윤곽 모델에 대해 초점을 둔다. 정의된 에너지 함수를 최소화하는 과정을 통해 객체의 윤곽을 추출하는 능동윤곽모델 (Active Contour Model; Snake)은 물체인식, 컴퓨터 및 로봇비전, 컴퓨터그래픽, 의료영상처리 분야에서 성공적으로 사용하고 있다[3-5]. 이 방법은 단순한 객체의 윤곽을 효율적으로 검출할 수 있는 장점이 있지만 많은 제약이 있다. 대표적인 제약은 초기 스네이크 포인트가 윤곽을 추출하려는 객체에 가깝게 설정하여야 하고, 객체 윤곽이 불규칙성이 없는 원형 모양을 이루는 단일 객체이어야 한다. 초기 스네이크 포인트 위치 및 객체의 불규칙성 문제를 해결하기 위해 많은 방법이 제안되었다. Xu[5]는 GVF(Gradient Vector Flow) 방법을 제안하였다. 오목한 윤곽을 추출하기 위해 외부에너지로 단순히 에지 정보를 이용하지 않고 에지의 공간 확산을 통해 얻은 GVF 에너지 맵을 이용해 객체의 오목한 윤곽까지 검출할 수 있었다. 그러나, GVF 에너지 맵

[†] 종신회원: 배재대학교 정보통신공학과
논문접수: 2010년 6월 7일
수정일: 1차 2010년 7월 13일, 2차 2010년 8월 16일
심사완료: 2010년 8월 29일

을 계산하는데 많은 시간이 필요하고, 호리병 모양의 윤곽을 갖는 객체에 대해서는 좁은 입구에서 GVF 에너지의 집중 현상으로 안쪽 윤곽까지 수렴하지 못한다. Kim과 Jang[6, 7]은 불규칙한 윤곽 및 호리병 모양의 객체 윤곽추출 문제를 해결하는 알고리즘을 제안하였다. 불규칙한 윤곽 추출을 위해 초기 스네이크 포인트가 객체 윤곽에 도달하면 2개의 스네이크 포인트를 연결하는 직선과 객체 윤곽의 거리를 계산하여 임계 값 보다 크면 스네이크 포인트를 증가하고 임계 값 보다 작으면 스네이크 포인트를 삭제하는 방법을 제안하였다. 호리병 모양의 객체 윤곽추출을 위해 Frenet Formula를 이용해 스네이크 포인트가 매우 오목한 윤곽까지 수렴하는 방안을 제안하였다. 그러나 이러한 방법들은 단일 객체에는 성공적으로 적용되지만 복수 객체에는 적용하지 못하는 한계점이 있다.

복수 객체의 윤곽을 추출하기 위해서는 Choi[8]는 스네이크를 분리하고 연결하는 알고리즘을 제안하였지만, 정의한 임계 값이 너무 많아 영상의 내용에 따라 성능이 결정된다. 그리고 에너지 함수를 반복 계산할 때 마다 스네이크 포인트를 증가함으로 소요 시간이 많이 걸리는 문제점이 있다. Velasco[9]는 Growing Snakes을 제안하여 복수 객체에 대한 윤곽추출을 하였지만 GVF에 에너지 맵 계산으로 많은 시간이 소요된다. 본 논문에서는 정의한 임계 값을 사용하지 않고, 에너지 함수를 반복할 때 마다 스네이크 포인트 수가 증가하지 않는 윤곽 검출 알고리즘을 제안한다.

논문의 구성은 다음과 같다. 먼저 2장에서는 외적 및 내적을 사용하여 초기에 복수 객체를 포함한 윤곽을 고속으로 분리하고 연결하여 각각이 윤곽이 단일 개체만을 포함하는 방법을 제안한다. 3장에서는 각각의 윤곽에 포함된 단일 객체의 윤곽을 추출하는 개선된 능동윤곽모델 알고리즘을 설명한다. 4장 및 5장에서는 실험결과 및 결론을 설명한다.

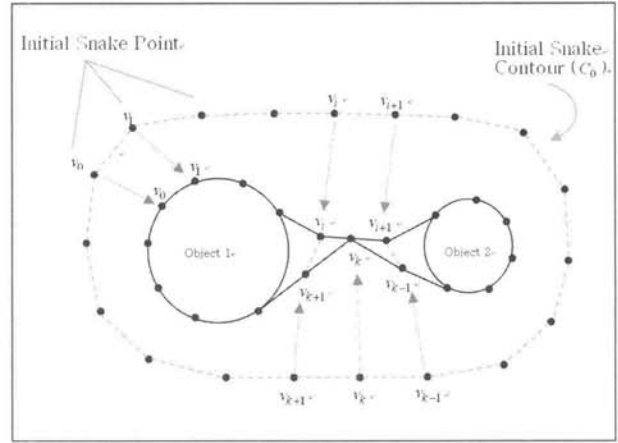
2. 고속 스네이크 분리 및 연결 알고리즘

(그림 1) (a)는 큰 원 과 작은 원 객체가 2개 있는 영상이다. 초기 스네이크 포인트를 순서적으로 연결하여 얻은 폐곡선은 초기 스네이크 윤곽(Initial Snake Contour, C_0)이다. 세그먼트 ($S_{i,i+1}$) 및 세그먼트 벡터 ($\overrightarrow{S_{i,i+1}}$)는 시작 포인트(v_i)와 끝 포인트(v_{i+1})을 연결하여 얻은 선 및 벡터이다. 세그먼트 단위 벡터 ($\overrightarrow{s_{i,i+1}}$)는 식 (1)에 정의한다.

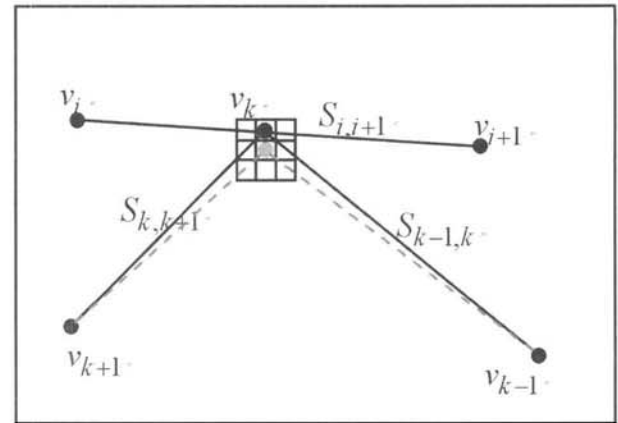
$$\overrightarrow{s_{i,i+1}} = \overrightarrow{S_{i,i+1}} / \| \overrightarrow{S_{i,i+1}} \| \quad (1)$$

여기에서 $\| \cdot \|$ 는 길이를 나타낸다.

에너지 함수를 최소화하는 반복 연산을 하면 스네이크 포인트가 객체의 윤곽 방향으로 이동하면서 스네이크 포인트 v_k 는 $S_{i,i+1}$ 와 교차하게 된다. (그림 1) (b)는 스네이크 포인트가 세그먼트를 교차하는 영역을 확대한 그림이다. 스네이크 포인트가 교차하면 객체 윤곽 추출에 필요한 스네이크 포



(a)



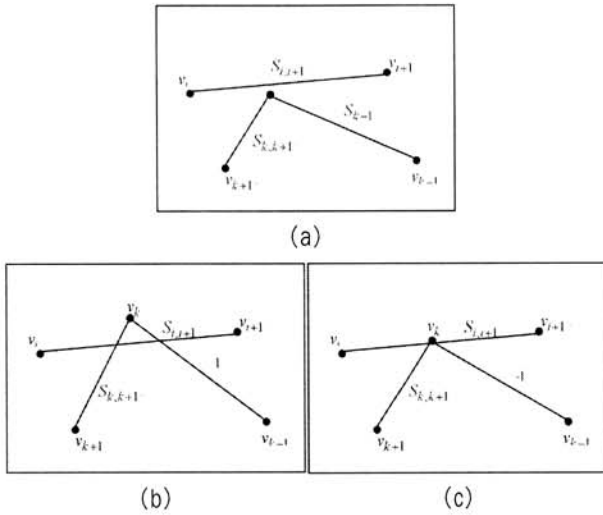
(b)

(그림 1) (a) 스네이크 포인트의 이동 과정 (b) 스네이크 세그먼트의 교차영역 확대

만 분리하여 연결하고, 필요 없는 스네이크 포인트는 삭제한다. 스네이크 포인트의 분리, 연결, 삭제 연산을 통해 만들어진 스네이크 윤곽의 수는 객체의 수와 같아야 하고, 단일 객체를 포함하는 각각의 스네이크 윤곽은 계속된 에너지 함수의 최소화하는 연산을 통해 윤곽 내에 있는 객체의 윤곽을 추출하여야 한다. 본 논문에서는 미리 정의한 임계 값 및 시간이 많이 소요되는 에지 맵을 이용하는 않는 고속 알고리즘을 제안한다.

2.1 세그먼트 교차를 결정하는 법

(그림 2)에 3개의 세그먼트 $S_{i,i+1}$, $S_{k-1,k}$, 및 $S_{k,k+1}$ 가 있다. (그림 2) (a)는 3개의 세그먼트가 교차하지 않는 경우이므로 스네이크 포인트를 분리, 연결, 삭제할 필요가 없다. 에너지 함수의 최소화하는 연산을 계속하면 스네이크 포인트가 객체의 윤곽 방향으로 이동한다. (그림 2)(b)와 (c)는 스네이크 포인트가 객체의 윤곽방향으로 이동하면서 세그먼트가 교차하거나 접촉한 경우이다. 이런 경우에는 불 필요한 스네이크 포인트는 제거하고, 객체의 윤곽 추출에 필요한 스네이크 포인트는 분리하고 연결한다. 이 연산을 하면 스네이크 포인트를 연결한 순서는 변경됨으로 스네이크 윤곽을 구성



(그림 2) 세그먼트의 교차 설명: (a) 교차하지 않음 (b) 교차 (c) 교차(접촉)

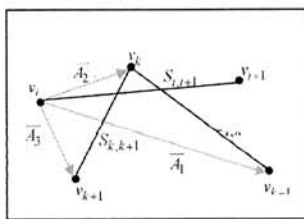
하고 있는 스네이크 포인트의 순서를 재 구성한다.

2.2 스네이크 포인트의 제거, 분리 및 연결

(그림 3)은 스네이크 포인트 제거에 대해 설명한다. v_i 와 v_{k-1} 를 연결한 벡터를 \vec{A}_1 과 단위벡터를 \vec{a}_1 으로 정의한다. v_i 와 v_k 를 연결한 벡터를 \vec{A}_2 이며 단위벡터는 \vec{a}_2 이다. v_i 와 v_{k+1} 를 연결한 벡터를 \vec{A}_3 이며 단위벡터는 \vec{a}_3 이다. $S_{i,i+1}$ 을 연장하면 평면이 2개의 영역으로 나누어진다. $S_{k-1,k}$ 와 $S_{k,k+1}$ 가 한쪽 영역에만 놓여 있으면 $S_{i,i+1}$ 와 교차하지 않고, 양쪽 영역 또는 접촉하면 교차하게 된다. 세그먼트가 한쪽 영역에 있는지 양 쪽 영역에 걸쳐 있는지를 결정하기 위해서는 $S_{i,i+1}$ 의 기준으로 외적 벡터 \vec{b}_k 인 식 (2)을 정의한다.

$$\vec{b}_k = \vec{s}_{i,i+1} \times \vec{a}_k \quad \text{for } k=1,2,3 \quad (2)$$

식 (2)는 $S_{k-1,k}$ 와 $S_{k,k+1}$ 가 한쪽 영역에 놓여 있으면 \vec{b}_k 의 z 성분의 부호는 같고, 그렇지 않으면 부호가 다르다. $S_{k-1,k}$ 와 $S_{k,k+1}$ 의 끝점이 $S_{i,i+1}$ 의 선에 있으면 \vec{b}_k 의 z 성분의 부호는 0이 된다. 교차한 스네이크 포인트의 \vec{b}_k 의 부호는 다르므로 불필요한 스네이크 포인트는 v_k 가 된 $\vec{s}_{k,k-1}$ 다. 스네이크 포인트를 제거하는 방법은 식 (3)에 정의한다.



(그림 3) 스네이크 포인트의 제거

$$\text{value} = (\vec{b}_1 \cdot \vec{b}_2 \leq 0 \wedge \vec{b}_2 \cdot \vec{b}_3 \leq 0) \quad (3)$$

여기에서 \bullet 는 내적을 설명한다.

value=1이면, v_k 를 제거하고, 그렇지 않으면 v_k 를 제거하지 않는다.

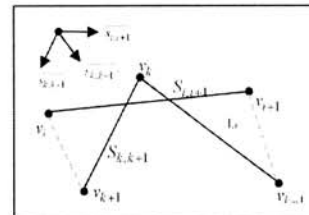
스네이크 포인트가 제거되면 앞과 뒤에 있는 스네이크 포인트는 순서적으로 연결되지 않고 분리된다. 분리된 스네이크 포인트는 객체의 윤곽을 추출할 수 있도록 적절하게 연결하여야 한다. 스네이크 포인트를 연결하는 기준은 (그림 4)에 같이 같은 쪽에 스네이크 포인트를 연결한다. 연결하는 식은 식 (4)에 정의한다.

$$\text{value} = \vec{s}_{i,i+2} \cdot \vec{s}_{k,k+1} < \vec{s}_{i,i+1} \cdot (-\vec{s}_{k-1,k}) \quad (4)$$

value = 1이면, v_i 를 v_{k+1} 에, v_{i+1} 를 v_{k-1} 에 각각 연결하고, 그렇지 않으면, v_i 를 v_{k-1} 에, v_{i+1} 를 v_{k+1} 에 연결한다. 분리 및 연결 연산 후에 스네이크 윤곽을 구성하는 스네이크 포인트의 순서는 식 (5)와 같이 재배열하여 새로운 스네이크 윤곽을 얻는다.

$$C_i = \{v_i^b, v_i^1, v_i^2, \dots, v_i^e \mid i=0,1,2, \dots, N-1\} \quad (5)$$

여기에서 v_i^b 및 v_i^e 는 C_i 의 시작 및 최종 스네이크 포인트이다. N 는 윤곽의 수이다.



[그림 4] 스네이크 포인트의 분리 및 연결

3. 복수객체 윤곽 추출을 위한 개선된 스네이크 알고리즘

Kim과 Jang[7]은 매우 불규칙한 윤곽을 갖는 객체 및 호리병 모양의 윤곽을 갖는 객체의 윤곽을 추출하는 알고리즘을 제안하였다. 초기 스네이크 포인트가 객체의 윤곽에 수렴한 후 스네이크 포인트를 증가하여 객체 윤곽을 정확하게 추출하게 하였고, 호리병 모양의 객체윤곽 추출 문제를 해결하기 위해 Frenet Formula를 이용해 스네이크 포인트가 매우 오목한 윤곽까지 수렴하는 방안을 성공적으로 제안하였다. 그러나 이 방법은 단일 객체에 성공적으로 적용하였고, 본 논문에서는 이 방법을 복수객체에 적용하는 알고리즘을 제안한다. 다음의 내용은 [6]에 자세하게 설명하였다.

스네이크 포인트를 이동하게 하는 에너지 함수는 식 (6)

과 같이 내부에너지(Internal Energy) $E_{int}(v_i)$ 와 외부에너지(External Energy) $E_{ext}(v_i)$ 의 합으로 나타낸다.

$$E_{snake}(v) = \sum_{i=0}^{M-1} (E_f(v_i) + E_{ext}(v_i)) \quad (6)$$

여기에서 $v_i = (x_i, y_i)$ 는 i 번째 스네이크 포인트이고, x_i 와 y_i 는 i 번째 스네이크 포인트의 x 와 y 의 좌표이다. M 은 전체 스네이크 포인트의 개수이다.

내부 에너지는 식 (7)와 같이 스네이크 포인트의 1차 미분과 2차 미분의 합으로 이루어 진다. 내부에너지의 최소화 하는 반복과정은 스네이크 포인트를 연결하는 곡선을 부드럽게 한다.

$$E_{int}(v_i) = \alpha \|v_i - v_{i-1}\|^2 + \beta \|v_{i-1} - 2v_i + v_{i+1}\|^2 \quad (7)$$

내부에너지의 하나인 1차 미분에서 얻은 Continuity에너지는 현재 스네이크 포인트 v_i 와 다음 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 오목한 윤곽을 갖는 객체에서는 성공하지 못한다. 오목한 윤곽에서는 스네이크 포인트간의 길이가 더 필요함으로 스네이크 포인트의 평균거리를 최소화하는 대신 v_i 와 v_{i-1} 의 차의 절대값과 v_i 와 v_{i+1} 의 차의 절대값의 차를 이용하여 스네이크 포인트를 안쪽 깊숙이 오목한 윤곽까지 이동하였다. 식 (8)에 정의하였다.

$$E_{con}(v_i) = \frac{|\|v_{i-1} - v_i\| - \|v_i - v_{i+1}\||}{con_{max}} \quad (8)$$

여기서 con_{max} 는 검색윈도우내의 최대값을 나타낸다.

2차 미분 항인 Curvature 에너지는 현재 스네이크 포인트와 이웃한 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 접선벡터(Tangent Vector) T , 법선벡터(Normal Vector) N , 중법선벡터(Binormal Vector) B 를 사용하여 포인트가 윤곽에 수렴하는 방법을 제안하였다. 접선벡터는 1차 미분으로 얻을 수 있다. v_i 에서 1차 미분은 $v_{i+1} - v_i$ 로 근사화한다. 식 (9)에 정의한다. 법선벡터는 접선벡터의 미분, 즉 2차 미분으로 얻을 수 있고 식 (10)에 정의한다. 중법선 벡터는 $B(v_i) = T(v_i) \times N(v_i)$ 로 정의되고, 접선 및 법선 벡터를 알면 중법선 벡터를 식 (11)을 이용하여 얻을 수 있다.

$$T(v_i) \approx v_{i+1} - v_i \quad (9)$$

$$N(v_i) \approx v_{i-1} - 2v_i + v_{i+1} \quad (10)$$

$$B(v_i) = T(v_i) \times N(v_i) = \begin{vmatrix} x_i & y_i \\ x_i^T & y_i^T \\ x_i^N & y_i^N \end{vmatrix} = (x_i^T y_i^N - y_i^T x_i^N) e_z \quad (11)$$

스네이크 포인트가 에너지 최소화 연산을 통해 객체의 윤곽 방향으로 이동하면서 최종적으로는 객체의 윤곽에 정지하여야 한다. 스네이크 포인트를 객체의 윤곽에 정지하기 위해서는 외부에너지를 사용하는데 본 논문에서는 영상의 에지 정보를 사용하여 윤곽의 에지에서 최소값을 갖도록 하여 스네이크 포인트를 객체의 윤곽에 정지시킨다. 식 (12)로 정의한다.

$$E_{ext}(v_i) = -\gamma |\nabla f(v_i)|^2 \quad (12)$$

본 논문에 사용할 개선된 전체 에너지함수는 식 (13)와 같이 정의하였다.

$$E_{snake}(v) = \sum_{i=0}^{M-1} (\alpha E_{con}(v_i) + \beta E_{cur}(v_i) + \gamma E_{ext}(v_i)) \quad (13)$$

매개변수는 α 는 스네이크 포인트의 간격과 오목한 윤곽에서 스네이크 포인트를 얼마나 깊게 이동시킬 수 있는지를 결정하는 변수이다. 매개변수 β 는 스네이크 포인트를 윤곽 방향으로 이동시키는 변수이고 γ 는 스네이크 포인트를 객체의 윤곽에 정지시키는 변수이다. 매개변수인 α , β 그리고 γ 의 가중치는 스네이크 포인트를 객체의 윤곽에 정지시키는 데 중요한 변수이다. γ 의 가중치를 α 및 β 보다 더 크게 설정하여 스네이크 포인트가 객체의 윤곽에 도달하면 정지하도록 하였다. 여기서 매개변수 $\alpha = 1.0$, $\beta = 1.0$ 그리고 $\gamma = 1.2$ 의 가중치를 설정하였다.

3.1 복수객체의 윤곽 추출의 개선된 알고리즘

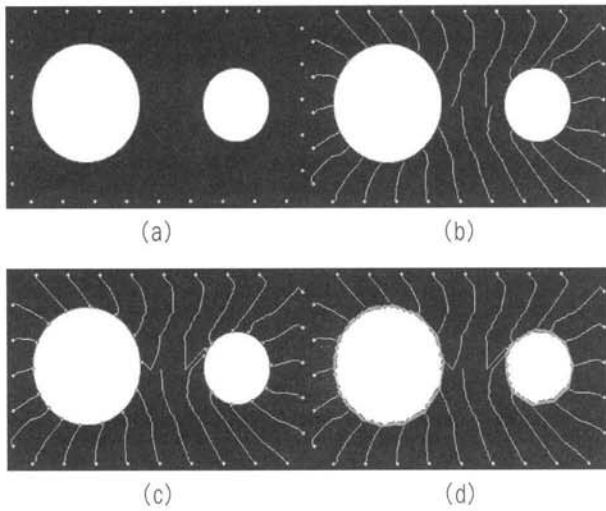
불규칙한 윤곽을 갖는 객체 및 호리병 모양의 윤곽을 갖는 단순 객체의 윤곽을 추출하는 알고리즘에 복수 객체 윤곽 추출을 적용하기 위해 알고리즘을 개선하였다. 알고리즘의 단계는 아래에 설명한다.

- 단계 1: 에너지함수를 계산하고 최종이면 끝, 그렇지 않으면 단계 2로 간다.
- 단계 2: 스네이크 포인트 제거 연산; 스네이크 포인트가 세그먼트와 교차 또는 접촉한지를 식 (3)을 이용하여 스네이크 포인트를 제거하고 단계 3으로 간다. 제거할 스네이크 포인트가 없으면 단계 1로 간다.
- 단계 3: 스네이크 포인트 분리 및 연결 연산; 식 (4)을 이용하여 스네이크 포인트를 분리하고 연결한다. 단계 4

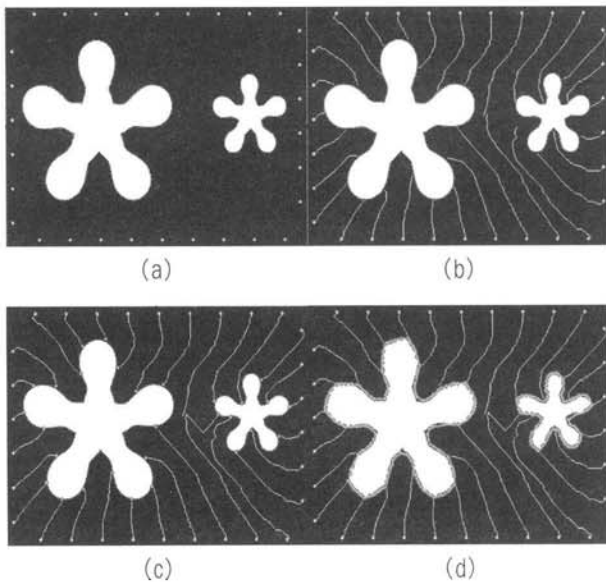
로 간다.

단계 4: 스네이크 포인트를 재 배열; 윤곽을 구성하는 스네이크 포인트 순서를 식 (5)을 이용하여 재 배열한다. 단계 1로 간다.

(그림 5)는 개선된 알고리즘을 적용하여 크기가 다르고 단순한 윤곽을 2 개의 원인 복수객체의 윤곽을 추출하는 과정을 단계적으로 설명하였고, (그림 6)은 불규칙한 윤곽을 가지고 있는 복수객체에 적용하였다.



(그림 5) 제안한 알고리즘을 적용하여 크기가 다른 2개의 원인 단순한 윤곽을 갖는 복수객체의 윤곽추출에 대한 단계적 과정 설명; (a) 초기 스네이크 윤곽 (b) 세그먼트 교차 (c) 스네이크 포인트의 제거, 분리, 및 연결 (d) 복수객체의 윤곽 추출



(그림 6) 2 개의 불규칙한 윤곽을 갖는 복수객체의 윤곽추출에 대한 단계적 과정 설명; (a) 초기 스네이크 윤곽 (b) 세그먼트 교차 (c) 스네이크 포인트의 제거, 분리, 및 연결 (d) 복수객체의 윤곽 추출

4. 실험결과

본 논문에서 제안한 객체 추출 방법은 펜티엄-IV 3.0GHz, 메모리 1Gbyte 환경에서 Visual C++ 6.0을 이용하여 구현하였다. 제안한 방법과 기존의 방법을 성능평가를 위해 Relative Shape Distortion (RSD)를 이용하여 예측된 윤곽선의 정확성을 정량화하여 비교하였다. RSD는 식 (14)에 정의한다.

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

여기서 R_{est} 는 예측된 객체 영역이 되고, R_{ori} 는 실제 객체의 영역이 된다. \oplus 는 이진 XOR 연산을 나타낸다.

(그림 7) (a)~(d)는 불규칙한 윤곽 및 복잡한 구조를 갖는 복수 객체의 가상영상이다. (그림 7) (c)와 (d)는 각각 폐와 도로에 있는 자동차의 복수객체의 자연영상이다. 성능비교를 위해 제안한 알고리즘과 대표적인 다른 2개의 알고리즘인 GVF 및 적응적인 힘[8]을 이용한 알고리즘을 구현하였다. (그림 7)의 파란색으로 표시한 궤도는 스네이크 포인트가 수렴한 궤도를 나타내고, 초록색 점은 최종 수렴한 위치를 표시한다. <표 1>은 성능비교를 보여준다. 제안한 방법은 GVF와 성능 면에서는 비슷하지만 GVF는 에너지 맵 계산으로 소요시간이 많이 걸리는 단점이 있다. 적응적 힘 방법은 에너지 함수 계산 시 적응도를 계산하여 스네이크 포인트를 추가한다. 스네이크 포인트가 추가되면 스네이크 포인트 마다 에너지 함수를 계산하므로 시간이 더 소요된다. 제안한 한 방법은 에너지 맵 및 적응도를 사용하지 않고 빠른 속도로 계산할 수 있는 내외적 벡터를 사용하여 속도를 개선하였다.

<표 1> Performance Comparison

		GVF	적응적 힘	제안한 알고리즘
a	Time	32.50s	1.344s	0.535s
	RSD	0.0010	0.0116	0.0045
b	Time	55.62s	2.450s	1.026s
	RSD	0.0010	0.0097	0.0038
c	Time	140s	4.334s	2.495s
	RSD	0.0030	0.2500	0.0032
d	Time	116s	4.786s	2.263s
	RSD	0.0030	0.1500	0.0040
e	Time	95.5s	2.860s	1.307s
	RSD	0.0100	0.0850	0.0290
f	Time	122s	5.350s	1.820s
	RSD	0.0200	0.1000	0.0320

	GVF	적응적 힘	제안한 알고리즘
a			
b			
c			
d			
e			
f			

(그림 7) 제안한 알고리즘과 2개의 다른 알고리즘의 복수객체 윤곽추출

5. 결론

본 논문은 매우 불규칙한 윤곽을 갖는 복수객체의 윤곽 추출 방법을 제안하였다. 복수객체의 윤곽추출을 할 때 각각의 객체 윤곽을 해당하는 스네이크를 빠르고 효율적으로 분리하는 방법이 매우 중요하다. 제안한 방법은 임계값 및 에너지 맵을 계산하지 않고 외적 및 내적 계산으로 스네이크를 고속으로 분리하는 방법을 제안하였다. 제안한 방법은 몇 배의 빠른 속도로 정확하게 복수객체의 윤곽을 추출하였다. 향후 연구내용으로는 복수 객체의 윤곽의 트래킹에 대해 연구가 필요하다.

참고 문헌

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

[4] 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.

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

[6] 김신형, 장중환, "스테레오 동영상에서 스네이크를 이용한 객체 윤곽 추적 알고리즘," *정보처리학회논문지 B 제12권-B권, 제7호*, 767-774쪽, 2005년.

[7] 김신형, 장중환, "오목한 윤곽을 갖는 객체에서 스네이크 기반의 윤곽선 검출 방법," *정보처리학회논문지B 제13-B권 제4호*, 361-368쪽, 2006년 8월.

[8] Wai-Pak Choi, Kin-Man Lam, Wan-Chi Siu, "An Adaptive Active Contour Model for Highly Irregular Boundaries," *Pattern Recognition*, Vol.34, pp.323-331, 2001.

[9] F. A. Velasco and J. L. Marroquin, "Growing Snakes: Active Contours for Complex Topologies," *Pattern Recognition*, Vol.36, pp.475-482, 2003.



장 중 환

e-mail : jangjw@pcu.ac.kr

1979년 한양대학교 전자통신공학과(공학사)

1986년 North Carolina 주립대학교 전기 컴퓨터공학과(공학석사)

1990년 North Carolina 주립대학교 전기 및 컴퓨터공학과(공학박사)

1990년~현 재 배재대학교 정보통신공학

과 교수

관심분야: 영상처리, 컴퓨터 비전