

복수객체의 윤곽추출을 위한 스네이크 분리 및 연결 알고리즘의 실험적 분석

Guo Cui[†] · 황재용^{**} · 장종환^{***}

요 약

복수객체의 윤곽추출을 위해 스네이크를 분리하고 연결하는 대표적인 방법이 스네이크 포인트의 거리를 이용한 최소거리방법이다. 이 방법은 객체 위상에 따라 스네이크를 분리하지 못하는 경우가 발생한다. 본 논문에서는 스네이크를 분리하지 못하는 경우를 실험적으로 증명하고 스네이크 세그먼트의 벡터를 이용한 새로운 방법이 스네이크를 성공적으로 분리 및 연결함을 실험적으로 보임으로써 최소거리방법의 문제점을 해결하였다. 하나의 영상 안에 3개 및 5개 객체가 있는 실험영상에 실험을 하여 제안한 방법이 우수하다는 것을 보여준다.

키워드 : 스네이크, 분리 및 연결, 객체윤곽추출, 복수객체

Experimental Analysis of Algorithms of Splitting and Connecting Snake for Extracting of the Boundary of Multiple Objects

Guo Cui[†] · JaeYong Hwang^{**} · JongWhan Jang^{***}

ABSTRACT

The most famous algorithm of splitting and connecting Snake for extracting the boundary of multiple objects is the nearest method using the distance between snake points. It often can't split and connect Snake due to object topology. In this paper, its problem was discussed experimentally. The new algorithm using vector between Snake segment is proposed in order to split and connect Snake with complicated topology of objects. It is shown by experiment of two test images with 3 and 5 objects that the proposed one works better than the nearest one.

Keywords : Snake, Splitting and Connecting, Object Contour Extraction, Multiple Objects

1. 서 론

스네이크(Snake)[1-4]를 이용한 복수객체의 윤곽을 추출하는 방법은 2단계로 구성한다. 첫 번째 단계는 각각의 객체를 포함 하도록 스네이크를 분리 및 연결하는 단계이고, 두 번째 단계는 각각의 스네이크 안에 포함 된 단일 객체의 윤곽을 추출하는 단계이다[5-6].

예를 들면, 하나의 초기 스네이크 내부에 N 개의 객체가 있는 경우, 초기 스네이크를 분리하고 연결하여 각각의 객체를 포함하도록 N개의 스네이크를 만든다. 초기 스네이크가 에너지 함수를 이용하여 객체 방향으로 이동하면 스네이

크가 교차한다. 본 논문에서는 첫 번째 단계인 스네이크 분리 및 연결에 대한 실험적인 연구이다.

스네이크 세그먼트(이하 세그먼트)가 교차 할 때 필요 없는 스네이크 포인트는 제거하고 필요한 스네이크 포인트는 분리하고 연결한다[7-9]. 대표적인 스네이크 분리 및 연결 방법은 스네이크 포인트의 거리를 이용하는 최소거리방법(the nearest method)이다[9]. 이 방법은 스네이크 포인트의 거리를 계산하여 필요 없는 스네이크 포인트는 제거하고 최소거리를 갖는 스네이크 포인트를 연결하는 방법이다. 이 방법의 문제는 최소거리를 갖는 스네이크 포인트를 연결하여 스네이크를 재 구성하면 객체 위상에 따라 스네이크가 분리되지 않는 경우가 발생한다. 본 논문에서는 이러한 경우를 실험적으로 증명하였고, 스네이크 세그먼트의 벡터를 이용하면 성공적으로 분리하고 연결하는 것을 실험적으로 보여 준다.

2장에서는 간단한 스네이크 분리 방법인 최소거리방법을

† 준 회원 : 배재대학교 정보통신공학과 석사과정
** 준 회원 : 배재대학교 정보통신공학과 박사수료
*** 종신회원 : 배재대학교 정보통신공학과 교수
논문접수 : 2012년 3월 20일
수정일 : 1차 2012년 5월 31일, 2차 2012년 7월 2일
심사완료 : 2012년 7월 4일
* Corresponding Author : JongWhan Jang(jangjw@pcu.ac.kr)

설명하고 3장에서는 벡터를 이용한 새로운 방법을 설명한다. 4장에서는 하나의 영상 내에 3개 및 5개의 객체가 있는 실험 영상에 최소거리 및 벡터 방법을 각각 적용하여 실험적으로 분리하지 못하는 경우를 증명하고 해결하였다. 마지막으로 향후 연구 방향과 결론을 논의한다.

2. 최소거리방법

스네이크는 에너지함수를 최소화하여 스네이크 포인트를 객체의 방향으로 이동한다. 하나의 스네이크 윤곽 안에 복수 객체를 포함하면 스네이크 포인트가 이동하면서 세그먼트가 그림 1과 같이 교차하게 된다. 세그먼트는 식 (1)로 주어진다.

$$y = a_i x + b_i \tag{1}$$

2개의 선형 식의 교차 판별은 x 및 y의 교차 점이 교차한 세그먼트의 공동 영역 안에 있어야 한다. 교차 점은 식 (2)로 주어진다.

$$x = \frac{\begin{vmatrix} -b_1 - 1 \\ -b_2 - 1 \end{vmatrix}}{\begin{vmatrix} a_1 - 1 \\ a_2 - 1 \end{vmatrix}} \quad y = \frac{\begin{vmatrix} a_1 - b_1 \\ a_2 - b_2 \end{vmatrix}}{\begin{vmatrix} a_1 - 1 \\ a_2 - 1 \end{vmatrix}} \tag{2}$$

교차한 세그먼트의 스네이크 포인트의 거리는 식 (3)로 주어진다.

$$d = |v_i - v_j| \text{ for } i = i, i+1 \text{ and } j = k-1, k+1 \tag{3}$$

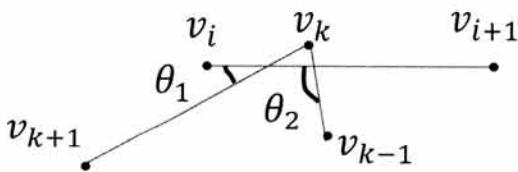


그림 1. 세그먼트의 교차; v_i 는 스네이크 포인트, S_{ij} 는 v_i 에서 v_j 를 연결한 세그먼트, θ_i 는 세그먼트 교차 각도
Fig. 1. Intersection of segment; v_i is a segment point, S_{ij} is a segment connected from v_i to v_j . and θ_i is an angle of segment intersection

여기에서 $| |$ 는 두 스네이크 포인트의 거리를 나타낸다.

식 (3)을 사용하여 4개의 스네이크 포인트의 거리를 계산 후 최소거리를 갖는 스네이크 포인트를 연결하는 방법이 최소거리방법이다. 그림 1에서 최소거리를 갖는 2개의 스네이크 포인트는 v_i 와 v_{k-1} 이다. 그러므로 v_i 와 v_{k-1} , v_{i+1} 와 v_{k+1} 을 연결하고 교차한 v_k 는 제거한다. 최소거리방법을 적용하여 새롭게 만들어진 스네이크는 그림 2에 굵은 선으로

구성한다. 굵은 선으로 연결된 세그먼트는 스네이크를 분리하지 못하고 다시 교차한다.

3. 벡터를 이용한 제안한 방법

그림 1에서 v_i 와 v_{i+1} 를 연결한 세그먼트는 $S_{i,i+1}$, 세그먼트 벡터는 $\vec{S}_{i,i+1}$, 세그먼트의 단위 벡터는 $\vec{S}_{i,i+1}$ 으로 정의한다. $S_{i,i+1}$ 을 연장하면 평면이 위와 아래의 2개의 영역으로 나눈다. $S_{k-1,k}$ 와 $S_{k,k+1}$ 가 한쪽 영역에 있으면 $S_{i,i+1}$ 와 교차하지 않고, 양쪽 영역 또는 접촉하면 교차한다. 세그먼트의 교차를 판별하기 위해 외적 벡터 \vec{a} 를 식 (4)로 정의한다.

$$\vec{a} = \vec{S}_{i,i+1} \times \vec{S}_{i,j} \text{ for } j = k-1, k, k+1 \tag{4}$$

세그먼트가 교차하면 해당하는 객체에 가까운 스네이크 포인트를 찾기 위해 세그먼트의 교차 각도를 계산하는 식 (5)로 정의한다.

$$\text{value} = \vec{S}_{i,i+1} \cdot \vec{S}_{k,k+1} < \vec{S}_{i,i+1} \cdot (-\vec{S}_{k-1,k}) \tag{5}$$

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. 제안한 방법을 이용하여 스네이크를 분리 및 연결하면 가는 점선 표시되어 스네이크를 성공적으로 분리한다.

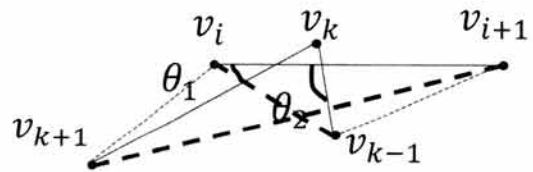


그림 2. 최소거리방법 및 제안한 방법을 적용한 스네이크 분리 및 연결 결과. 굵은 점선 및 가는 점선은 각각 최소거리방법 및 제안한 방법을 적용한 후 얻은 새로운 세그먼트
Fig. 2. The result of dividing and connecting snake by using both the nearest method and the proposed method. The bold dashed line and thin dotted line are new segments by applying the nearest method and the new proposed method respectively

4. 실험 결과

본 논문의 실험환경은 펜티엄-IV 3.0GHz, 메모리 1Gbyte 환경에서 Visual C++ 6.0을 사용하여 구현하였다. 실험 영상의 x 및 y의 좌표는 왼쪽 위가 (0, 0), 오른쪽 아래가 (256, 256)이다. 초기 스네이크 윤곽은 복수객체를 포함하는 사각형으로 구성하고 윤곽을 구성하는 스네이크 포인트의 거리는 같은 간격으로 나눈다. 스네이크 포인트는 에너지함수의

최소화를 통해 객체방향으로 이동시킨다. 스네이크 포인트의 이동 경로는 영상 내에 빨간 색의 실선으로 표시한다. 그림 3은 초기 스네이크 윤곽 내에 3개의 객체를 포함하는 실험영상이다. 에너지함수의 최소화를 이용하여 스네이크 윤곽이 객체방향으로 이동하면서 세그먼트가 교차한다.

교차한 세그먼트의 좌표는 흰색 세그먼트는 시작 점이 (94, 114), 끝 점이 (97, 87)이다. 긴 청색 세그먼트는 시작 점이 (112, 70), 끝 점이 (97, 95), 짧은 세그먼트는 시작 점이 (97, 95), 끝 점이 (104, 95)이다. 교차점은 (97, 95)이다. 최소 거리방법을 사용하여 스네이크 포인트의 거리를 계산하면 $d_{i,k-1} = \sqrt{2161}, d_{i,k+1} = \sqrt{130}, d_{i+1,k-1} = \sqrt{541}, d_{i+1,k+1} = \sqrt{113}$ 이다. 스네이크 포인트의 최소거리는 v_{i+1} 와 v_{k+1} 의 거리인 $\sqrt{113}$ 이다. 그러므로 v_k 를 제거하고, v_{i+1} 를 v_{k+1} 에, v_i 를 v_{k-1} 에 각각 연결하면 그림 4(a) 같이 스네이크는 분리하지 못한다. 제안한 방법을 사용하기 위해 세그먼트 벡터를 계산한다.

$\overrightarrow{S_{i,i+1}} = (97,87) - (97,114) = (0,-27)$ 이고, $\overrightarrow{S_{i,i+1}} = (0,-1)$ 이다. $-\overrightarrow{S_{k,k-1}} = (112,70) - (97,95) = (15,-25)$ 이고, $-\overrightarrow{S_{k,k-1}} = (0.51,-0.85)$ 이다. $\overrightarrow{S_{k,k+1}} = (104,95) - (97,95) = (7,0)$ 이고, $\overrightarrow{S_{k,k+1}} = (1,0)$ 이다.

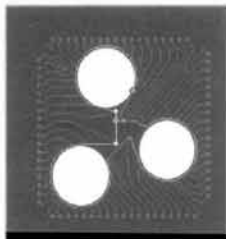


그림 3. 초기 스네이크 윤곽 내에 3개의 객체가 있는 실험영상. 스네이크 윤곽이 에너지 함수의 최소화를 통해 객체 방향으로 이동하면서 세그먼트가 교차함을 보여주는 실험영상 Fig. 3. Test image with 3 objects within the initial contour. Segment intersection is occurred when snake contour is moving toward objects by minimizing energy function

$\overrightarrow{S_{i,i+1}} \cdot \overrightarrow{S_{k,k+1}} = (0,-1) \cdot (1,0) = 0$ 이고, $\overrightarrow{S_{i,i+1}} \cdot -\overrightarrow{S_{k,k-1}} = (0,-1) \cdot (0.51,-0.58) = 0.85$ 이다. 식 (5)에 주어진 value 값이 1이 되어 v_k 를 제거하고 v_i 는 v_{k+1} 에, v_{i+1} 는 v_{k-1} 에 연결하면 그림 4(b) 같이 스네이크는 교차하지 않고 성공적으로 분리된다.

두 번째 실험은 초기 스네이크 윤곽 안에 5개의 객체를 포함한 실험영상에 최소거리방법 및 제안한 방법을 적용하였다. 최종결과영상을 얻을 때까지 4번의 세그먼트가 교차가 일어난다. 표 1은 세그먼트 교차할 때의 스네이크 포인트의 좌표이다. 그림 5는 세그먼트 교차가 일어날 때 스네이크 분리 및 연결을 보여준다. 첫 번째 및 두 번째 세그먼트가 교차할 때 최소거리방법 및 제안한 방법은 성공적으로 스네이크를 분리한다. 그러나 그림 5(c)와 같이 3번째 교차

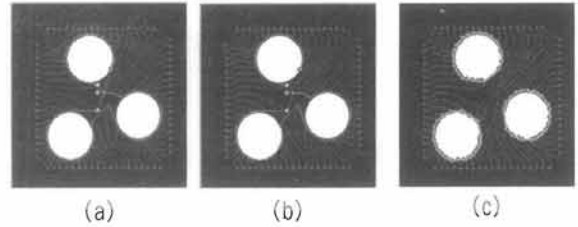


그림 4. 세그먼트가 교차할 때 스네이크 분리 및 연결; (a) 최소거리방법; 세그먼트를 분리하지 못하고 객체의 윤곽을 추출할 수 없음. (b) 벡터를 사용한 제안한 방법; 세그먼트를 분리함. (c) 제안한 방법을 사용하여 3개의 객체의 윤곽을 추출한 최종영상.

Fig. 4. Division and connection of snake: (a) the nearest method can't divide segment and extract object contour. (b) the proposed method can divide segment and extract object contour. (c) the final image successfully extracted object contour by using the proposed method

가 일어날 때 최소거리방법은 스네이크를 분리하지 못하여 객체의 윤곽을 성공적으로 추출하지 못 한다. 그러나 제안한 방법을 사용하면 스네이크를 분리 및 연결하여 그림 5(c)와 같이 5개의 객체의 윤곽을 성공적으로 추출할 수 있다. 표 2는 최소거리방법과 벡터를 이용한 새로운 방법에 대한 성능비교이다.

표 1. 5개의 객체를 포함하는 실험영상에서 세그먼트가 교차할 때 스네이크 포인트의 좌표

Table 1. Location of snake point at segment intersection in test image with 5 objects

	v_i	v_{i+1}	v_{k-1}
the first segment intersection	(77,75)	(57,79)	(79,52)
the second segment intersection	(125,145)	(118,120)	(119,108)
the third segment intersection	(111,87)	(111,105)	(95,66)
the fourth segment intersection	(67,141)	(75,137)	(78,128)

	v_k	v_{k+1}
the first segment intersection	(74,76)	(98,77)
the second segment intersection	(119,124)	(0,16)
the third segment intersection	(111,95)	(112,59)
the fourth segment intersection	(73,138)	(75,150)

표 2. 최소거리방법과 벡터를 이용한 방법의 성능비교 Table 2. Performance comparison of the nearest method and the proposed method

	accuracy of contour extracted	computation time
the nearest method	10.500	2.45s
the proposed method	0.150	2.50s

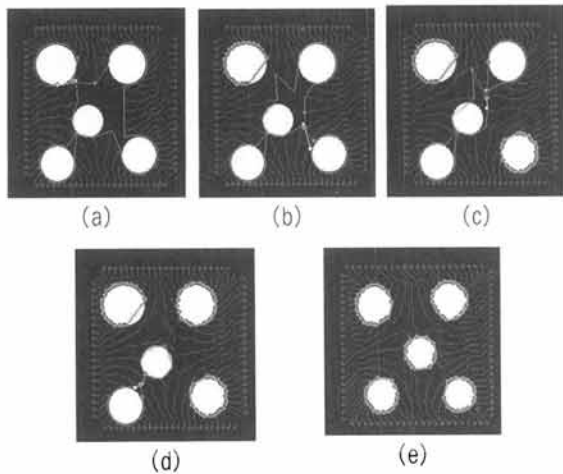


그림 5. (a) 첫 번째 세그먼트 교차 (b) 2번째 세그먼트 교차 (c) 3번째 세그먼트 교차 (d) 4번째 세그먼트 교차 (e) 최종결과영상

Figure 5. (a) the first segment intersection (b) the second segment intersection (c) the third segment intersection (d) the fourth segment intersection (e) the final result image

5. 결 론

본 논문에서는 대표적인 스네이크 분리 및 연결 방법인 최소거리방법에 대한 문제를 실험적으로 논의하였고, 해결 방안으로 벡터를 이용한 알고리즘을 제안하였다. 벡터를 이용한 방법은 복수 객체에 포함하는 스네이크를 성공적으로 분리하여 복수객체의 윤곽을 추출하였다. 스네이크 방법은 곡선전계 방식보다는 계산량이 적다는 장점이 있지만 위상 처리에 한계점이 있다. 제안한 방법은 어느 정도 객체 위상 변화에 잘 적용되지만 앞으로 연구방향은 곡선전계 방식과 성능을 비교하는 것이 중요하다.

참 고 문 헌

[1] M. Kass, A. Witkin and D. Terzopoulos, "Snake: active contour models," International Journal of Computer Vision, Vol.1, No.4, pp.321-331, 1987.
 [2] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," IEEE Transaction on Image Processing, Vol.7, No.3, pp.359-369, 1998.
 [3] Wai-Pak Chai, Kin-Man Lam, and Wan-Chi Siu, "An adaptive active contour model for highly irregular boundaries," Pattern Recognition, Vol.34, pp.323-331, 2001.
 [4] S. H. Kim, A. Alatter, and J. W. Jang. "Snake-based contour detection for objects with boundary concavities," Optical Engineering, Vol.47, No.3, pp.037002-1 - 037002-7, March, 2008.

[5] JongWhan Jang, "Active Contour Model for Boundary Detection of Multiple Objects," The KIPS Transactions: Part B, Vol.17-B, No.5, pp.375-380, Oct., 2010.
 [6] C. S. Tong, P. C. Yuen, and Y. Y. Wong, "Snake algorithm for multiple object segmentation," Optical Engineering, Vol.41, No.12, pp.3177-3182, Dec., 2002.
 [7] L. Li, J. Liu, and M. Fox. "Segmentation of external force field for automatic initialization and splitting of snakes," Pattern Recognition. Vol.38, Issue 11, pp.1947-1960, Nov., 2005.
 [8] Shoichi Araki, "Splitting of active contour models based on crossing detection for extraction of multiple objects," Systems and Computers in Japan, Vol.28, No.11, pp.34-42, 1997.
 [9] Miki Haseyama and Yukinori Yokoyama, "Moving object extraction using a shape-constraint-based splitting active contour model," 2005 12th International Conference on Image Processing (ICIP), 2005.



Guo Cui

e-mail : lovecuiquo@163.com

2007년 7월 천젠공업대학교(공학사)

2010년 8월~현 재 배재대학교 정보통신

공학과 석사과정

관심분야: 영상처리



황재용

e-mail : platans@hanafos.com

1997년 배재대학교 정보통신공학과(공학사)

2004년 배재대학교 정보통신공학과

(공학석사)

2007년 배재대학교 정보통신공학과

(박사수료)

관심분야: 영상처리, 컴퓨터 네트워크



장종환

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~현 재 배재대학교 정보통신공학과 교수

관심분야: 영상처리, 컴퓨터 비전