

물체의 윤곽선 추출을 위한 진동 스네이크

윤진성[†]·김관중^{††}·김계영^{†††}·백두원^{††††}

요약

스네이크라 불리는 능동적 윤곽선 모델은 물체 윤곽선 추출에 유용한 도구이다. 그러나 전형적인 스네이크는 많은 계산시간을 요구하며, 에너지 함수의 특성에 의하여 복잡한 모양의 윤곽선을 추출하지 못하는 경우가 있다. 또한 초기 스네이크의 위치와 형태에 매우 의존적이다. 본 논문에서 상기의 문제들을 해결하기 위하여 탐욕적 알고리즘에 기반한 진동 스네이크를 제안한다. 진동 스네이크는 두 단계로 구성된다. 첫 번째 단계에서는 스네이크를 이루는 정점들을 적합성에 따라 직접 윤곽선으로 이동시킨다. 두 번째 단계에서는 지역적 최소를 방지하기 위하여 일부 정점들을 정렬한다. 이러한 과정에 의해 스네이크가 진동하는 모습을 보여준다. 실험에서 진동 스네이크의 진동과정과 탐욕적 알고리즘 간의 성능비교를 보여준다. 진동 스네이크는 복잡한 모양의 윤곽선을 정확하게 추출하면서 탐욕적 알고리즘에 비해 대략 5배정도 속도가 향상되었다.

A Shaking Snake for Contour Extraction of an Object

Jin-Sung Yoon[†] · Kwan-Jung Kim^{††} · Gye-Young Kim^{†††} · Doo-Won Paik^{††††}

ABSTRACT

An active contour model called snake is powerful tool for object contour extraction. But, conventional snakes require exhaustive computing time, sometimes can't extract complex shape contours due to the properties of energy function, and are also heavily dependent on the position and the shape of an initial snake. To solving these problems, we propose in this paper an improved snake called "shaking snake," based on a greedy algorithm. A shaking snake consist of two steps. According to their appropriateness, we in the first step move each points directly to locations where contours are likely to be located. In the second step, we then align some snake points with a tolerable bound in order to prevent local minima. These processes shake the proposed snake. In the experimental results, we show the process of shaking the proposed snake and comparable performance with a greedy snake. The proposed snake can extract complex shape contours very accurately and run fast, approximately by the factor of five times, than a greedy snake.

키워드: 진동 스네이크(Shaking Snake), 에너지 최소화 알고리즘(Energy Minimization Algorithm), 윈도우 확장(Extending Window), 정렬(Alignment)

1. 서론

영상에서 물체의 윤곽선은 인식, 분류, 추적 등 컴퓨터 비전 분야에 사용되는 중요한 특징이다. 이러한 윤곽선을 추출하기 위해 스네이크라 불리는 활동적 윤곽선 모델이 여러 분야에서 연구 및 사용되고 있다.

스네이크는 정점의 집합으로 이루어진 곡선으로써 관심 물체의 주변에 초기화된 후, 에너지 함수와 에너지 최소화 알고리즘에 의해 물체의 윤곽선으로 변형되는 기능을 지닌 모델이다. Kass에 의해 처음 제안된 스네이크 모델의 에너지 최소화 알고리즘은 오일러(euler) 방정식을 이용한 행렬

연산에 기반하여 수행된다[1]. 이 알고리즘은 고차미분 방정식을 이용하기 때문에 수학적으로 불안정하며 에너지 함수의 특성상 스네이크가 움츠러드는 문제점이 있다. 수학적인 불안정을 보완하기 위해 Amini는 동적 프로그래밍에 기반한 에너지 최소화 알고리즘을 제안하였다. 또한 스네이크를 이루는 정점들로 하여금 일정한 거리를 유지하게 하거나, 높은 에지 강도를 가지는 화소들로만 수렴하도록 하는 등의 강한 제약(hard constraint)에 대해 소개하였다[2]. 이 알고리즘은 스네이크의 수학적 불안정성을 해결하였지만, 재귀적 처리로 인하여 많은 계산시간이 소요되는 단점이 있다. Ray는 스네이크의 수행속도를 향상시키기 위해 Kass의 행렬연산 기법을 개선하였으며, 다해상도 영상을 이용한 점진적 윤곽선 추출 기법을 제안하였다[3]. Williams는 에너지 최소화 알고리즘의 복잡성을 간편화하고 수행속도를 향상시키기 위해 탐욕적(greedy) 알고리즘에 기반한 에너지

* 본 연구는 한국과학재단 목적기초연구(ROI-2202-000-00561-0(2202)) 지원으로 수행되었음.

† 준회원: 숭실대학교 대학원 컴퓨터학과

†† 정회원: 한서대학교 컴퓨터정보학과 교수

††† 중신회원: 숭실대학교 컴퓨터학부 교수

†††† 정회원: 숭실대학교 미디어학부 교수

논문접수: 2003년 5월 16일, 심사완료: 2003년 8월 8일

최소화 방법을 제안하였다[4]. 또한 여러 가지 곡률(curvature) 계산법에 대해 연구하였으며, 곡률에 따라 에너지 함수에 대한 가중치들을 가변적으로 조절함으로써 정점들이 높은 곡률을 가진 꼭지점으로 수렴하도록 하였다. Lam은 탐욕적 스네이크의 속도를 더욱 향상시키기 위해 에너지 최소화점을 찾기 위한 탐색범위인 여덟 개의 주변화소 모두를 고려하는 대신 네 개의 주변 화소만을 선별적으로 고려하는 방법을 제안하였다[5]. Yan은 윤곽선과 스네이크간의 거리를 고려하여 정점들을 빠르고 정확하게 윤곽선으로 이동시키고자 피드백(feedback) 에너지 함수와 전역적 예지 탐지자를 개발하였다[6].

많은 연구를 통하여 스네이크가 처음 개발되었을 때 가지던 단점들이 다수 해결되었지만 다음과 같은 문제들이 남아 있다. 에너지 최소화 알고리즘이 많은 수에 걸쳐 반복 수행되며, 에너지 최소화점을 찾기 위해 불필요한 공간을 탐색하기 때문에 수행 시간의 낭비를 감수해야 한다. 내부 에너지 함수의 특성에 의해 스네이크는 형태 변형에 제약을 받아 복잡한 모양의 윤곽선을 추출하지 못한다. 또한 초기 스네이크의 위치와 모양에 대해 높은 의존성을 지니기 때문에 관심 물체의 주위에 스네이크가 초기화되지 않았을 때 올바르게 동작하지 못하는 경우가 발생한다.

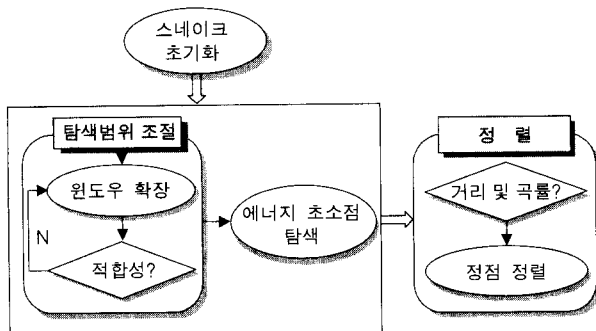
본 논문에서는 이러한 문제들을 해결하기 위해 개선된 에너지 최소화 알고리즘을 제안한다. 제안된 알고리즘은 탐욕적 알고리즘에 기반을 두고 있으며, 스네이크를 이루는 정점들의 적합성을 조사하여 적합하지 않은 정점의 윈도우를 점진적으로 확장시켜 에너지 최소화점을 찾기 위한 탐색범위를 넓히는 방법을 사용한다. 즉, 에너지를 최소화 하는 위치를 찾기 위한 탐색범위를 정점 주변의 직사각형으로 국한하지 않고 가변적으로 조절함으로써 복잡한 모양의 윤곽선을 빠르고 정확하게 수렴하도록 한다. 현재 처리중인 정점의 적합성만 고려하여 윈도우를 확장하기 때문에 경우에 따라서 정점이 지역적 최소점(local minima)에 빠질 수 있다. 때문에 스네이크의 연속성과 완만성을 보존하면서 지역적 최소점 문제를 해결하기 위해 알고리즘의 매 반복 후에 정점들을 정렬(alignment)시키는 방법을 제안한다.

(그림 1)은 본 논문에서 제안하는 진동 스네이크의 전체적인 구조를 보여주고 있다. (그림 1)에서 보는 바와 같이 진동 스네이크는 크게 탐색범위를 조절하는 단계와 정점을 정렬하는 단계로 구성된다. 탐색범위를 조절하는 단계에서는 정점의 적합성 여부에 따라 점진적으로 윈도우를 확장한다. 그 후 확장된 윈도우를 이용하여 에너지를 최소화 하는 위치를 찾아 정점을 이동시킨다. 모든 정점에 이러한 과정을 적용한 후, 정점들에 대한 거리와 곡률을 계산하여 임계값을 만족하지 않는 정점들은 이전 정점과 이후 정점의 중간위치로 이동시켜 정렬시킨다. 알고리즘이 종료될 때까지 이러한 과정을 반복 수행한다. 본 논문에서는 제안하는 진동 스네이크의 효율성을 증명하기 위해 다양한 영상에 대한 실험결과를 보인다. 또한 탐욕적(greedy) 스네이크 알고리즘과 진동 스네이크 알고리즘의 성능을 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 사용되는 에너지 함수를 설명하고, 3장에서는 진동 스네이크 알고리즘에 대해 기술한다. 4장에서 실험 결과를 5장에서 결론을 맺는다.

2. 윤곽선 추출을 위한 에너지 함수

스네이크는 정점들의 집합으로 이루어진 곡선으로써 에너지 함수와 에너지 최소화 알고리즘에 의해 윤곽선으로 수렴하며 전체 윤곽선으로 수렴하였을 때 에너지가 최소화된다. 알고리즘은 이동된 정점의 수가 일정 수 이하일 때 종료된다. Kass가 제안한 에너지 함수는 식 (1)과 같다[1]. 식 (1)의 첫 번째 항은 내부 에너지 함수로써 식 (2)와 같이 두 개의 항으로 구성된다. 식 (2)의 각 항은 스네이크의 연속성과 완만성을 제어하는 함수이다. 즉, 스네이크를 이루는 정점들 간의 간격이 작고 완만한 곡선을 이룰수록 적은 값을 가진다. 여기서 α, β 는 각 항의 상대적 중요성을 고려한 가중치이다. 식 (1)의 두 번째 항은 식 (3)과 같으며, 영상 특징에 기인하는 외부 에너지 함수로써 스네이크를 이루는 정점을 물체의 윤곽선으로 이동시키는 역할을 한다. 식 (1)의 세 번째 항은 외부로부터 주어지는 제약사항으로써 인터페이스를 통해 스네이크를 영상 내 원하는 위치로 초기화시키거나 정점이 임의 위치로 이동하는 것을 제한하는 등의 제약사항을 의미한다.



(그림 1) 진동 스네이크의 전체 구조도

$$E_{snake}^* = \int_0^1 E_{snake}(v(s)) ds \tag{1}$$

$$= \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)) ds$$

$$s \in [0, 1]$$

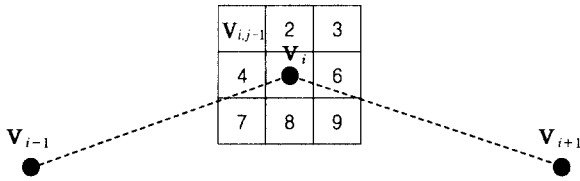
$$v(s) = (x(s), y(s))$$

$$E_{int} = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2 \tag{2}$$

$$E_{image} = -|\nabla I(v)|^2 \tag{3}$$

Kass가 제안한 에너지 함수는 다음과 같은 문제점들이 있다. 첫째, 연속성에 대한 함수는 정점들 간의 거리가 적을수록 적은 값을 가지므로 스네이크가 움츠러들게 한다. 둘째, 완만성에 대한 함수는 스네이크가 완만한 곡선을 이룰수록 작은 값을 나타내기 때문에 스네이크가 높은 곡률을 지닌 물체의 꼭지점으로 수렴하지 못하게 한다.

본 논문에서는 사용하는 에너지 함수와 에너지 최소화 알고리즘은 Williams가 제안한 탐욕적 스네이크에 기반한다 [4]. 탐욕적 스네이크는 (그림 2)와 같이 각 정점 i 에 윈도우를 설정하고 윈도우 내에서 에너지가 가장 최소화되는 위치 j 를 찾아 정점을 이동시키는 에너지 최소화 알고리즘을 반복 수행함으로써 윤곽선을 추출한다.



(그림 2) 에너지 최소화점 탐색 윈도우

이때 계산되는 에너지 함수는 식 (4)와 같으며 이산 공간 상에서 계산이 이루어지기 때문에 각 에너지 함수의 매개 변수는 영상 내에서의 각 정점의 좌표값이 된다. 식 (4)의 첫 번째 항은 연속성을 제어하는 내부 에너지 함수로써 식 (5)와 같다.

$$E_{snake}^* = \sum_{i=1}^N (\alpha E_{cont}(\mathbf{v}_i) + \beta E_{curve}(\mathbf{v}_i) + \gamma E_{image}(\mathbf{v}_i)) ds \quad (4)$$

$$\mathbf{v}_i = (x_i, y_i)$$

$$E_{cont}(\mathbf{v}_i) = \frac{|d^* - \|\mathbf{v}_i - \mathbf{v}_{i-1}\||}{\max_j \{|d^* - \|\mathbf{v}_i - \mathbf{v}_{i,j}\||\}}, \quad j=1 \dots 8 \quad (5)$$

여기서 i 는 현재 처리 중에 있는 정점을, j 는 정점 i 자신과 여덟 개의 이웃 화소를 나타낸다. d^* 는 모든 정점들 간의 평균거리이다. d^* 를 거리 측정에 이용하므로 스네이크를 이루는 정점들 간의 간격이 균일할수록 작은 값을 나타낸다. 즉, 스네이크가 움츠러드는 현상을 방지한다. 식 (4)의 두 번째 항은 스네이크가 이루는 곡선의 완만성을 제어하는 함수로써 식 (6)과 같다.

$$E_{curve}(\mathbf{v}_i) = \frac{|\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}|^2}{\max_j \{|\mathbf{v}_{i-1} - 2\mathbf{v}_{i,j} + \mathbf{v}_{i+1}|^2\}}, \quad j=1 \dots 8 \quad (6)$$

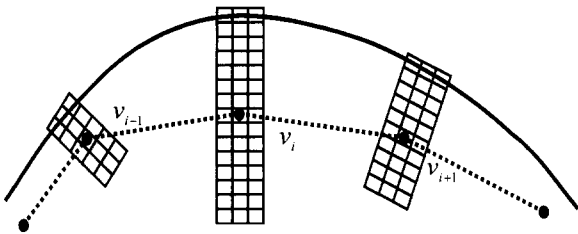
식 (3)의 세 번째 항은 영상 특징에 대한 함수로써 식 (7)과 같이 나타낸다. 식 (7)에서 mag 는 기울기 강도를 나타내며, 소벨(sobel) 연산자를 사용하였다. \max 와 \min 은 주변 화소들 중에서 최대, 최소 기울기 강도를 나타낸다. 에너지 함수들은 모두 1에서 0의 값을 지니도록 정규화된다.

$$E_{edge}(\mathbf{v}_i) = \frac{\min - mag(\mathbf{v}_i)}{\max - \min} \quad (7)$$

3. 탐욕적 알고리즘에 기반한 진동 스네이크

탐욕적 스네이크 알고리즘은 현재 처리중인 정점의 윈도우 내에서 에너지를 최소화 하는 위치를 찾아 스네이크의 에너지가 전역적으로 최소화 되도록 하는 알고리즘이다[4]. 이 알고리즘은 정점이 한번에 이동할 수 있는 범위가 윈도우 내로 한정되어 있기 때문에 다소의 연산시간의 낭비를 감수해야 하며 초기 스네이크 위치에 따라 예상치 못한 결과를 얻을 수 있다. 이것은 윈도우의 크기에 기인하는 문제로써 윤곽선 추출에 중요한 변수가 된다. 예를 들어 윈도우의 크기가 작고 윤곽선과 스네이크간의 거리가 멀면 정점이 윤곽선에 도달하기까지 많은 수의 반복 계산이 요구된다. 반대로 윈도우의 크기가 크면 곡률이 비교적 큰 모서리 같은 윤곽선을 수렴하지 못하며, 정점이 윤곽선을 지나치는 경우가 발생한다. 때문에 적절한 윈도우 크기를 결정하는 것은 중요하면서도 어려운 문제이다. Pardo와 Fu는 탐색 공간으로 인한 속도문제를 극복하기 위해 현재 정점의 이전 및 이후 정점들이 이루는 직각 방향으로 확장된 윈도우를 사용하였다[7,8]. 그러나 내부 에너지 함수와 고정된 윈도우로 인해 정점이 윤곽선을 지나칠 수 있으며 탐색공간이 확장함에 따라 에너지 함수 계산에 따르는 수행시간 낭비 등의 문제가 남아 있다.

이 문제를 해결하기 위하여 본 논문에서는 각 정점의 적합성을 조사하고 적합하지 않은 정점의 윈도우 크기를 점진적으로 확장하는 방법을 제안한다. 적합성은 현재 정점의 윈도우 내에 임계 값보다 높은 기울기 강도($\text{gradient magnitude}$)를 가지는 화소의 존재 유무에 의하여 결정된다. 만약 정점이 적합하면 즉, 윈도우 내에 임계 값보다 높은 기울기 강도를 가지는 화소가 존재하면 윈도우 내에서 에너지가 최소화되는 위치를 찾는다. 반대로 정점이 적합하지 않으면 적당한 방향으로 일정 거리만큼 윈도우를 확장시키고 다시 적합성을 조사한다. 이때 또 다시 정점이 적합하지 않는 것으로 판정되면 이동했던 방향의 반대방향으로 일정 거리 만큼 윈도우를 확장시킨다. 이러한 과정을 반복하여 정점이 적합한 것으로 판정될 때까지 윈도우를 점진적으로 확장시킨다. 윈도우가 확장되면서 적합한 것으로 판정되면 확장을 멈추고 확장된 윈도우 내에서 에너지가 최소인 위치를 찾아 정점을 이동시킨다. (그림 3)은 윈도우가 확장되는 방향과 형태를 나타낸 것이다. 정점을 이어주고 있는 실선은 스네이크를, 굵은 선은 윤곽선을 의미하며, 각 정점의 윈도우는 식 (8)에 의해 계산된 방향과 그 반대 방향으로 확장된다. 식 (8)은 현재 처리중인 정점에서 Δ 만큼의 이전의 정점과 Δ 만큼의 이후의 정점이 이루는 각도를 의미한다.



(그림 3) 가변적 윈도우 확장

$$\theta(v_i) = \tan^{-1} \left(\frac{y_{i+d} - y_{i-d}}{x_{i+d} - x_{i-d}} \right) \quad (8)$$

이때 에너지 함수는 식 (9)와 같다. 식 (9)에서 i 는 현재 처리중인 정점을 의미하며, m 은 윈도우 크기로 정점의 적 합성에 따라 증가할 수 있다. 확장된 윈도우에서 에너지 최소점을 찾을 때에는 각 에너지 함수에 적용되는 가중치 중 α 와 β 의 값은 0에, γ 의 값은 1에 가까운 값으로 설정한다. 윈도우를 확장시켜 정점을 윤곽선으로 이동시킬 때에는 스네이크의 연속성과 완만성은 거의 의미가 없어지기 때문이다. 즉, 확장된 윈도우의 끝 부분에 위치한 화소가 실제 윤곽선을 이루는 화소일 가능성이 높지만 연속성과 완만성에 대한 에너지 함수가 높은 수치를 나타내어 에너지 최소점으로 선택되지 못하는 경우를 방지하기 위해서이다.

$$E_{snake} = \sum_{j=1}^m \min_j \{ \alpha E_{cont}(v_{i,j}) + \beta E_{curv}(v_{i,j}) + \gamma E_{image}(v_{i,j}) \}, \quad (9)$$

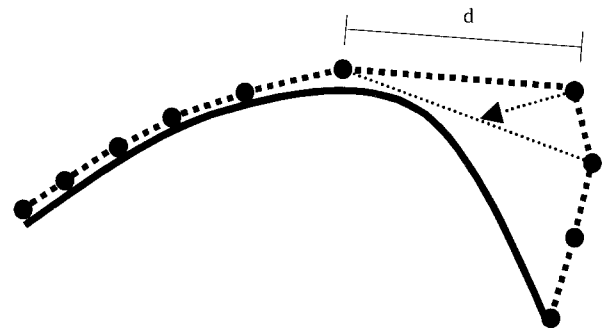
$j = 1 \dots m$

탐욕적 스네이크 알고리즘은 고정된 윈도우 내에서만 에너지를 계산하지만 개선된 알고리즘은 윈도우를 점진적으로 확장시킨다. 그 이유는 스네이크를 이루는 정점과 실제 윤곽선간의 거리가 클 때 해당 정점의 윈도우 내 화소들은 거의 모두 낮은 에지 강도를 나타내지만 반대로 윤곽선에 인접해 있는 경우 높은 에지 강도를 나타내는 특성을 이용하기 위해서이다. 즉, 윤곽선이 있을 가능성이 높은 곳으로 탐색공간을 점진적으로 확장하여 전체 탐색시간을 줄이고 정확도를 높이고자 하는 것이 목적이다. 윈도우가 확장될수록 탐색범위는 증가하지만 윤곽선을 추출하기 위한 알고리즘의 반복 수는 감소하기 때문에 전체적인 수행속도가 향상된다.

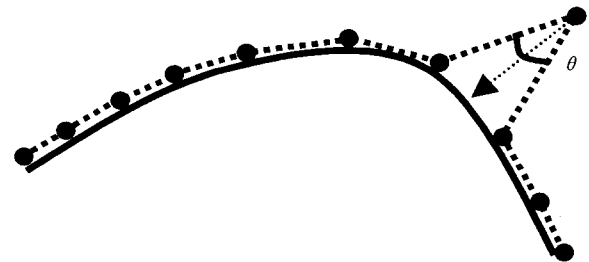
수행속도를 더욱 향상시키기 위해 확장된 윈도우 내에 존재하는 모든 화소들을 고려할 필요가 없다. 윈도우를 확장시키는 이유는 임계값보다 높은 기울기 강도를 가지는 화소의 위치로 직접 정점을 이동시키기 위해서이다. 때문에 본 논문의 실험에서는 윈도우가 확장되는 양방향에서 일정 수 이상의 임계값보다 큰 기울기 강도를 가지는 화소가 발견될 때까지 윈도우를 확장시킨 후, 발견된 화소들만을 고려하여 에너지 최소점을 찾는다. 이때 적용되는 각 에너지

함수에 대한 가중치는 모두 1로 설정한다. 양방향의 화소들 중에서 스네이크의 연속성과 완만성에 보다 자연스러운 형태를 지니는 화소를 선택하기 위해서이다.

윈도우를 확장시켜 정점을 윤곽선으로 이동시키는 방법은 잡음에 민감하다. 즉, 지역적 최소점에 빠질 가능성이 높다. 때문에 알고리즘의 매 반복 후, 각 정점들간의 거리와 곡률을 조사한다. 만약 그 거리가 정점들간의 평균거리보다 일정한 배수 이상 크거나 곡률이 임계값 보다 크다면 그 정점을 잡음으로 간주하여 이전 정점과 이후 정점의 중간 위치로 강제 이동시킨다. 잡음에 빠진 정점은 주변 정점들과 비약적으로 멀리 떨어져 있거나 비정상적으로 낮은 곡률을 가진다는 특성을 이용한 것으로 강한 제약사항(hard constraint)으로 작용한다.



(그림 4) 거리를 이용한 정렬의 예



(그림 5) 곡률을 이용한 정렬의 예

(그림 4)와 (그림 5)는 이러한 잡음의 특성을 이용한 정렬과정을 보여주고 있다. 점선과 굵은 선이 각각 스네이크와 윤곽선을 의미한다. (그림 4)는 다른 정점들과 비정상적으로 멀리 떨어져 있는 정점이 잡음으로 간주되어 이전 정점과 이후 정점의 중간 위치로 이동되는 과정을 보여주고 있으며, (그림 5)는 다른 정점과 비정상적으로 낮은 곡률을 이루고 있는 정점이 이전 정점과 이후 정점의 중간위치로 이동되는 과정을 보여주고 있다. (그림 4)의 경우 정렬된 정점이 여전히 윤곽선과 떨어져 있지만 알고리즘의 반복 수행을 통해 결국 윤곽선으로 이동된다.

식 (6)에서 v_{i-1} 에서 v_i 로, v_i 에서 v_{i+1} 로 이어지는 두 곡선이 이루는 곡률은 각 곡선의 길이에 영향을 받는다. 즉, 두 곡선의 길이가 같은 경우보다 틀린 경우에 값이 크다.

예를 들어 직선 상에 세 점이 놓여져 있을 때, 두 곡선의 길이가 다르면 식 (6)의 값은 0이 되지 않는다. 따라서 정렬과정에서 각 정점들의 곡률을 계산할 때 식 (10)을 이용한다. 식 (10)의 경우, 곡선의 길이를 정규화하기 때문에 두 곡선의 길이 차로 인한 곡률 계산상의 문제가 없어진다[4].

$$\Delta(v_i) = \left| \frac{\mathbf{u}_i}{|\mathbf{u}_i|} - \frac{\mathbf{u}_{i+1}}{|\mathbf{u}_{i+1}|} \right|^2 \quad (10)$$

$$\mathbf{u}_i = (x_i - x_{i+1}, y_i - y_{i+1})$$

$$|\mathbf{u}_i| = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

<표 1>은 본 논문에서 제안하는 진동 스네이크를 알고리즘 형태로 나타낸 것이다. 먼저 현재 처리 중에 있는 정점의 윈도우에 임계값보다 큰 에지 강도를 나타내는 화소가 존재하는지 조사한다. 존재한다면 윈도우 내에서 에너지가 최소인 위치를 찾는다. 그렇지 않으면 이 조건이 만족할 때까지 윈도우의 크기를 점진적으로 확장한다. 그 후 확장된 윈도우 내에서 에너지가 최소인 위치를 찾아 정점을 이동시킨다. 알고리즘의 매 반복 후에 거리와 곡률을 이용하여 일부 정점들을 정렬시킨다. 윈도우 확장과 정렬과정의 반복 수행에 의하여 스네이크의 정점들이 진동하는 모습을 보여준다. 진동 스네이크의 에너지 최소화 알고리즘은 평형(equilibrium) 상태를 이루었을 때 종료된다. 그러나 스네이크의 에너지는 이산 공간에서 계산되기 때문에 경우에 따라 몇몇 정점들이 진동(oscillation) 현상을 일으켜 평형상태에 도달하지 못하는 경우가 발생한다. 따라서 적절한 조건을 설정하여 종료를 유도하여야 한다. 본 논문에서의 에너지 최소화 알고리즘은 이동한 정점의 수가 임계값보다 적을 때 종료한다.

<표 1> 진동 스네이크 알고리즘

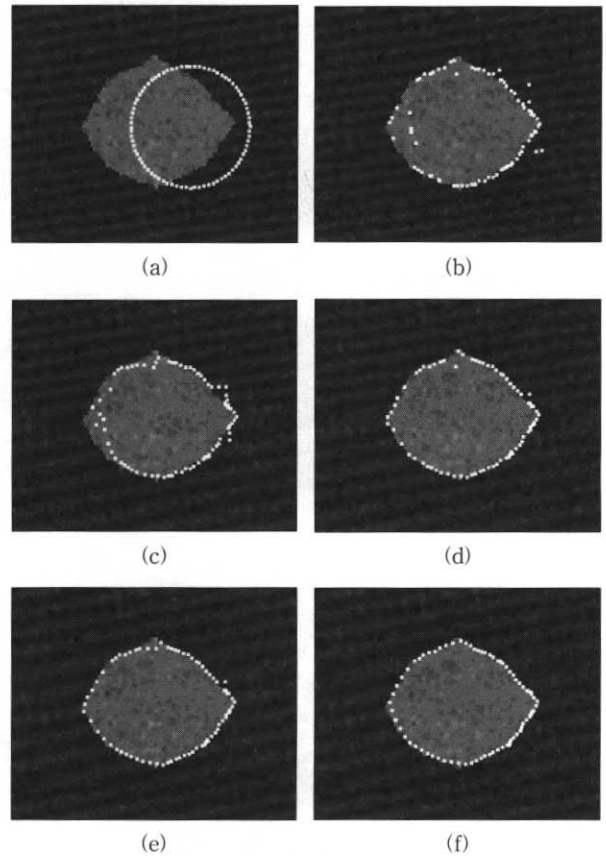
```

/* m denotes the size of a neighborhood
/* dir denotes the designated direction
Do
for i = 1 to n
    E_min = BIG
    m_i = 3 by 3
    if m_i is judged to be inappropriate then
        do extend m_i in dir_i and opposite dir_i
        until m_i is judged to be appropriate
        for j = 1 to m_i /* or to only the number of strong edge
            in m_i
                E_snake(v_j) = alpha E_cont(v_j) + beta E_curve(v_j) + gamma E_edge(v_j)
                if E_snake(v_j) < E_min then
                    E_min = E_snake(v_j),
                    j_min = j
        move point v_i to location of j_min
for i = 0 to n
    if (distance_i > th_distance) or (curvature_i > th_curvature) then
        move point v_i to middle location between v_i and v_{i-1}
until moved_points < th_moved_points
    
```

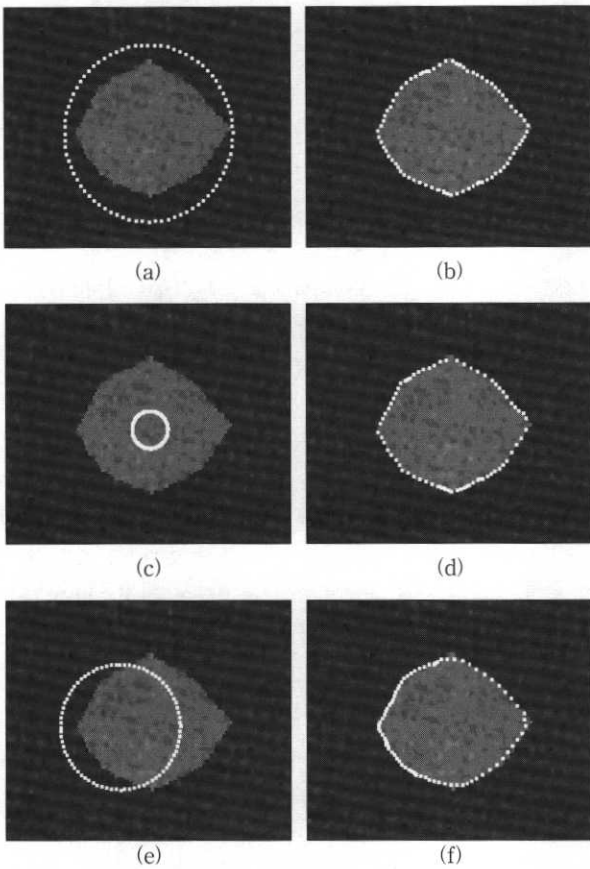
4. 실험 결과

본 장에서는 진동 스네이크를 다양한 종류의 영상에 적용한 실험결과를 보인다. 실험을 위해 사용한 컴퓨터는 IBM PC 펜티엄-4이며, 운영체제는 Microsoft Windows 2000을 구현 언어는 Visual C++ 6.0을 사용하였다. 알고리즘이 시작할 때 영상 화소의 평균 기울기 강도를 10으로 나누어 정점의 적합성 조사를 위한 임계값으로 사용하였다.

(그림 6)은 진동 스네이크가 윤곽선으로 수렴하는 과정을 보여준다. (그림 6)(a)는 스네이크의 초기 위치를, (그림 6)(b)는 윈도우 확장에 의해 스네이크의 정점들이 윤곽선으로 직접 이동된 결과를, (그림 6)(c)는 일부 정점들이 정렬된 후의 스네이크를 보여주고 있다. 나머지 (그림 6)(d)~(그림 6)(f)는 이후 확장과 정렬을 반복 수행하여 스네이크가 윤곽선으로 완전히 수렴하는 과정을 보여주고 있다. 윈도우 확장과 정점 정렬에 의해 스네이크의 정점들이 진동하면서 빠르고 정확하게 윤곽선으로 수렴하는 과정을 확인할 수 있다. 알고리즘이 반복될수록 윤곽선으로 수렴한 정점들의 수가 증가하며, 스네이크의 진동 폭이 작아진다. 즉, 알고리즘이 전역적 최소점(global minimum)에 근접할수록 진동하는 정점의 수와 진동의 폭이 작아지며, 진동 스네이크의 이러한 행위는 일부, simulated annealing 알고리즘과 유사하다[9].



(그림 6) 진동에 의한 윤곽선 수렴과정

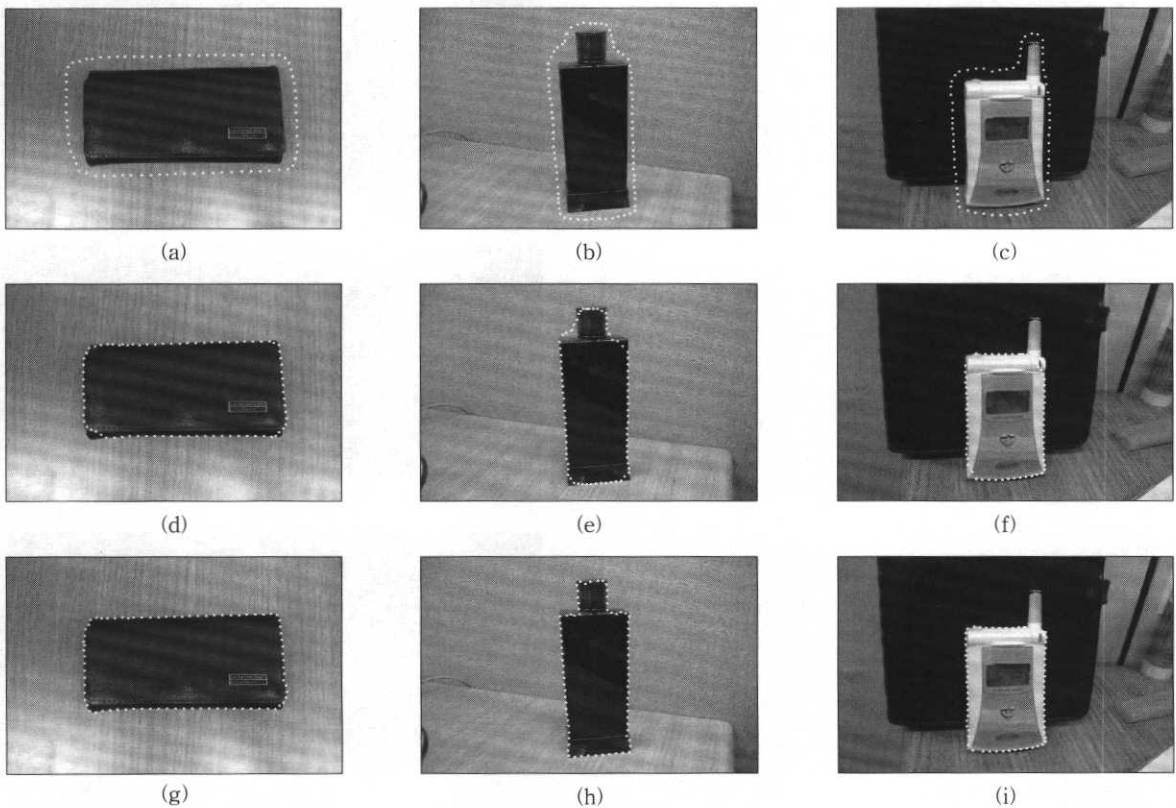


(그림 7) 초기 스네이크의 위치에 따른 결과

(그림 7)은 스네이크를 다른 위치에 초기화시켰을 때 윤곽선을 추출한 결과를 보여준다. (그림 7)(a)와 (그림 7)(c) 그리고 (그림 7)(e)는 스네이크가 초기화된 모습을 보여주며, (그림 7)(b)와 (그림 7)(d) 그리고 (그림 7)(f)는 그에 따라 윤곽선을 추출한 결과를 보여준다. 초기 스네이크의 위치가 다르더라도 결과는 거의 동일하다. 이것은 진동 스네이크가 초기 스네이크 위치에 강건함을 의미한다.

탐욕적 스네이크 알고리즘의 복잡도는 $O(nm)$ 으로써 가장 간결한 알고리즘으로 알려져 있으며 여러 참고 문헌에서 성능 비교를 위한 참고 모델로 사용되었다[5, 6, 10, 11]. (그림 8)의 영상들은 탐욕적 스네이크 알고리즘과 진동 스네이크 알고리즘의 성능 비교를 위해 사용된 실험영상으로써 모두 300×400 의 크기이다. 스네이크 초기화는 마우스를 이용하여 (그림 8)(a), (그림 8)(b), (그림 8)(c)와 같이 이루어졌으며, (그림 8)(d), (그림 8)(e), (그림 8)(f)와 (그림 8)(g), (그림 8)(h), (그림 8)(i)가 각각 탐욕적 스네이크와 진동 스네이크를 사용하여 윤곽선을 추출한 결과를 보여준다.

<표 2>는 (그림 8)에 대한 각 알고리즘의 반복횟수와 속도를 비교한 결과를 보여준다. <표 2>에서 TH1과 TH2는 각각 정렬과정에 사용된 거리와 곡률에 대한 임계값이다. 여러 번의 실험을 통해 정점들간의 평균 거리보다 3에서 5 배 이상의 거리나 135도에서 90도 이하의 곡률을 가지는 정점이 잡음일 가능성이 높음을 알 수 있었다. 따라서 TH1은



(그림 8) (a~c) 초기 스네이크, (d~f) 탐욕적 스네이크의 윤곽선 추출결과, (g~i) 제안하는 진동 스네이크의 윤곽선 추출결과

3(배)에서 5(배)로, TH2는 1에서 2로 설정하였다. 진동 스네이크 알고리즘이 탐욕적 스네이크 알고리즘에 비해 반복 횟수가 적고 속도가 빠르며 매 반복에 소요되는 시간 또한 적다. 이는 탐욕적 알고리즘이 정점을 이동시키는 과정에서 많은 교환작업을 수행하기 때문이다.

<표 3>은 (그림 8)에 대한 각 알고리즘이 추출한 윤곽선과 실제 윤곽선간의 평균 및 최대 거리를 보여준다. 실제 윤곽선은 마우스를 이용하여 실험자가 임의로 지정하였다. 제안하는 진동 스네이크 알고리즘은 윈도우 확장과 정렬에 의해 정점들을 직접 윤곽선으로 이동시키기 때문에 탐욕적 스네이크 알고리즘 보다 더욱 정확하게 윤곽선을 추출하며 특히 향수병을 추출한 결과처럼 폐곡선을 수렴할 수 있다.

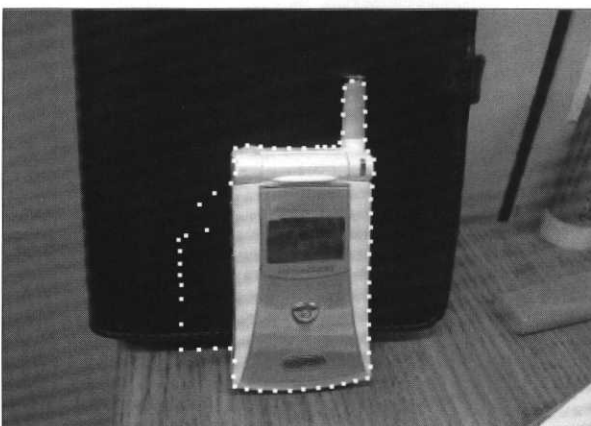
TH2에 큰 값을 적용하면 핸드폰의 안테나 부분 같은 큰 곡률을 가지는 윤곽선까지 수렴할 수 있지만, (그림 9)와 같이 몇몇 정점들이 지역적 최소점에 빠지는 경우가 발생할 수 있다. TH2에 2.0의 값을 적용 즉, 정점이 이루는 곡률을 90도까지 허용하였을 때의 결과이다.

<표 2> 속도 비교

영 상	정 점 수	탐욕적 알고리즘[4]		진동 알고리즘			
		반 복 횟 수	속 도 (sec)	TH1	TH2	반 복 횟 수	속 도 (sec)
지 갑	79	28	0.385	3	1.0	5	0.046
향수병	78	31	0.426	5	2.0	7	0.078
핸드폰	69	75	0.896	5	1.0	24	0.218

<표 3> 정확도 비교

영 상	탐욕적 알고리즘[4]		진동 알고리즘	
	평균거리	최대거리	평균거리	최대거리
지 갑	2.2	11.3	1.5	2.5
향수병	5.7	12.5	2.7	8.6
핸드폰	3.3	10.5	2.5	5.3



(그림 9) 지역적 최소점의 예

5. 결론 및 향후 연구

본 논문에서는 물체의 윤곽선을 빠르고 정확하게 추출하기 위해 에너지 최소화 알고리즘을 개선한 진동 스네이크를 제안하였다. 제안하는 진동 스네이크는 정점의 적합성을 조사하여 적합하지 않은 정점의 윈도우를 윤곽선이 존재할 것으로 예상되는 곳으로 점진적으로 확장시켜 정점을 직접 윤곽선으로 이동시킨다. 또한 알고리즘의 매 반복마다 정렬 과정을 통해 정점들이 지역적 최소점 빠지는 것을 방지하였다. 제안하는 진동 스네이크는 윈도우 확장에 의해 정점을 직접 윤곽선으로 이동시키기 때문에 수행속도가 빠르며 물체의 윤곽선을 정확하게 추출하였다. 또한 초기 스네이크 위치에 큰 영향을 받지 않는다.

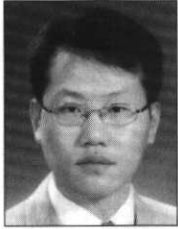
향후 진동 스네이크를 복잡한 배경에서의 이동물체 추적에 적용하고자 한다. 이때 이동물체 영역과 배경 영역을 분리할 수 있는 방법과 폐색이 일어났을 경우 폐색을 측정할 수 있는 방법이 필요하다. 향후 이를 위한 연구를 수행하고자 한다.

참 고 문 헌

- [1] Michael Kass, Andrew Witkin and Demetri Terzopoulos, "Snakes : Active Contour Models," International Journal of Computer Vision, Vol.1, No.4, pp.321-331, 1987.
- [2] Amir. A. Amini, Terry. E. Weymouth and Ramesh. C Jain, "Using Dynamic Programming for Solving Variational Problems in Vision," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.12, No.9, pp.855-867, 1990.
- [3] Nilanjan Ray, Bhabatosh Chanda, Jyotirmay Das, "A Fast and Flexible Multiresolution Snake with a Definite Termination Criterion," Pattern Recognition, Vol.34, No.7, pp. 1483-1490, 2001.
- [4] Donna J. Williams and Mubarak Shah, "A Fast Algorithm for Active Contours and Curvature Estimation," CVGIP : Image Understanding, Vol.55, No.1, pp.14-26, 1992.
- [5] K. M. Lam, H. Yan, "Fast Greedy Algorithm for Active Contours," Electronics Letters Vol.30, No.1, pp.21-23, January, 1994.
- [6] Lilian Ji, Hong Yan, "Attractable Snakes Based on the Greedy Algorithm for Contour Extraction," Pattern Recognition, Vol.35, No.4, pp.791-806, 2002.
- [7] X. M. Pardo, M. J. Carreira, A. Mosquera, D. Cabello, "A Snake for CT Image Segmentation Integrating Region and Edge Information," Image and Vision Computing, Vol.19, No.7, pp.461-475, May 2001.
- [8] Yue Fu, A. Tanju Erdem, A. Murat Tekalp, "Tracking Visible Boundary of Objects Using Occlusion Adaptive Motion Snake" IEEE Transaction On Image Processing, Vol.9, No.12 pp.2051-2060, December 2000
- [9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," Science, Vol.220, pp.671-680, 1983.
- [10] Wen-Nung Lie, Cheng-Hung Chuang, "Fast and Accurate Snake Model for Object Contour Detection," Electronics

Letters Vol.37, No.10, pp.624-626, May, 2001.

[11] Chun Leung Lam, Shiu Yin Yuen, "An Unbiased Active Contour Algorithm for Object Tracking," Pattern Recognition Letters, Vol.19, pp.491-498, April, 1998



윤진성

e-mail : jsyoon@vision.ssu.ac.kr

1995년 군산대학교 전자계산학과(이학사)
1997년 숭실대학교 전자계산학과(공학석사)
2002년~현재 숭실대학교 컴퓨터학과 박사
과정

관심분야 : 영상처리, 비디오 검색, 동작
이해 등



김관중

e-mail : kimkj@hanseo.ac.kr

1983년 숭실대학교 전자계산학과(공학사)
1988년 숭실대학교 컴퓨터학과(공학석사)
1998년 숭실대학교 컴퓨터학과(공학박사)
1997년~현재 한서대학교 컴퓨터정보학과
조교수

관심분야 : 컴퓨터구조, 마이크로프로세서, 병렬처리, VLSI 설계



김계영

e-mail : gykim@computing.ssu.ac.kr.

1990년 숭실대학교 전자계산학과(공학사)
1992년 숭실대학교 컴퓨터학과(공학석사)
1996년 숭실대학교 컴퓨터학과(공학박사)
1996년~1997년 한국전자통신 연구소
연구원

1997년~2001년 한국전력공사 전력연구원

2001년~현재 숭실대학교 컴퓨터학부 조교수

관심분야 : 컴퓨터 비전, 형태 인식, 생체 인식, VR/AR, 영상
처리 등



백두원

e-mail : dpaik@computing.ssu.ac.kr.

1983년 서울대학교 수학과(이학사)
1990년 미국 University of Minnesota
전산학(공학석사)
1991년 미국 University of Minnesota
전산학(공학박사)

1992년~1994년 미국 AT&T Bell Lab. MTS 연구원

1995년~현재 숭실대학교 미디어학부 조교수

관심분야 : 디지털 방송, 그래픽스, 알고리즘 등