*Journal of Korean*
*Data & Information Science Society*
*2007, Vol. 18, No. 2, pp. 439 ~ 445*

# Analyses of Computation Time on Snakes and Gradient Vector Flow

**Young-Tae Kwak[1)]**

**Abstract**

GVF can solve two difficulties with Snakes that are on setting initial contour and have a hard time processing into boundary concavities. But GVF takes much longer computation time than the existing Snakes because of their edge map and partial derivatives. Therefore this paper analyzed the computation time between GVF and Snakes. As a simulation result, both algorithms took almost similar computation time in simple image. In real images, GVF took about two times computation than Snakes.

***Keywords*** : Edge Map, Gradient Vector Flow, Snakes

## 1. 서론

오늘날 우리는 디지털 영상에 사용자의 의도를 반영하기 위하여 관심 영역을 추출하거나 영상의 일부를 변형하여 사용하고 있다. 이와 같은 작업에는 관심 영역내에 있는 물체의 윤곽선을 추출하는 기술이 기본적으로 포함된다. 윤곽선 추출은 영상처리의 기반기술로써 그 결과가 최종 결과에 크게 영향을 미치기 때문에, 영상처리나 패턴인식 문제에서 매우 중요하다.

일반적으로 간단한 윤곽선 추출 방법으로는 마스크를 이용하는 방법이 있다(장동혁(2003)). 마스크를 이용한 방법은 영상의 제1, 2차 미분을 이용하고 3×3 또는 5×5의 마스크를 이용하기 때문에 구현이 쉽고 수행 속도가 빠르다. 그러나 노이즈에 약하며 불필요한 윤곽선을 제거해야 하는 후처리 문제가 있다. 이외에도 텍스쳐 기반 윤곽선 추출 방법과 구조적인 접근 방법 등이 있다(Petrou(1999)).

마스크, 텍스쳐, 구조적 윤곽선 추출 방법은 임계치 조정외에 사용자의 의도를 반영할 수 없다. 이런 한계를 극복하기 위하여 Kass(1987)는 스네이크(Snakes)라는 active contour model를 제안하였다. 스네이크는 윤곽선의 모양을 결정하는 내부함수와 영상

1) 전북 익산시 마동 194-5 익산대학 컴퓨터정보과 조교수
E-mail : kwak@iksan.ac.kr

의 특징(윤곽선, 선 등)을 결정하는 외부함수로 구성된 에너지 함수를 만들고, 이런 에너지 함수를 최소화함으로써 영상의 특징으로 윤곽선을 추출하는 모델이다. 스네이크란 용어 또한 윤곽선의 추출과정이 동적으로 움직이는 뱀의 모양에서 유래했다.

Kass의 스네이크는 초기 정점의 위치 결정과 오목한 부분의 윤곽선 추출이 어렵다는 문제점이 있다. 이러한 문제의 해결 방법으로써, 초기 정점의 위치 결정은 곽영태(2006)가 볼록 다각형 근사화 방법에서 초기 정점을 물체 주위에 볼록 다각형으로 근사화 하는 방법을 제안하였고, 오목 부분 윤곽선 추출 문제는 Xu(1998)가 GVF (Gradient Vector Flow)를 제안하였다. GVF는 오목 부분 윤곽선 추출을 성공적으로 해결하지만, 영상으로부터 에지맵(edge map)을 만들고 GVF의 편미분 계산에서 상당히 많은 시간을 요구한다. 특히, 2차원 영상보다 3차원 영상인 경우는 더 그러하다.

따라서 본 논문에서는 실험을 통하여 스네이크와 GVF의 계산시간을 비교하여, 사용자가 자신의 실험 영상에 따라 스네이크나 GVF를 최적으로 선택할 수 있는 기준을 제안하고자 한다. 본 논문의 구성은 2장에서는 Kass의 스네이크 알고리즘을 간략하게 소개하고 3장에서 Xu의 GVF를 설명한다. 4장에서는 기초 영상과 실제 영상을 대상으로 실험하여 계산시간을 분석하고 스네이크나 GVF의 선택 기준을 제시하며 5장에서 결론을 맺는다.

## 2. 스네이크 알고리즘

스네이크는 2차원 영상 평면에서 스플라인 곡선 위의 정점들로 표현되며, 식(1)과 같이 정의된다.

$$\Omega = [0,1] \Rightarrow R^2 \qquad v(s) = (x(s), y(s)) \tag{1}$$

스네이크는 자신에게 작용하는 총에너지를 최소화 하는 방향으로 움직이는데, 여기에 작용하는 총에너지 $E^*_{snake}(v)$는 내부에너지와 외부에너지로 구성된다. 내부에너지는 스네이크의 형태에 영향을 주고, 외부에너지는 영상의 특징들로 끌어당기는 역할을 한다.

$$\begin{aligned} E^*_{snake}(v) &= \int_0^1 E_{snake}(v(s))ds \\ &= \int_0^1 (E_{int}(v(s)) + E_{ext}(v(s)))ds \end{aligned} \tag{2}$$

여기서 내부에너지는 식(3)과 같이 정의되며, 정점들이 서로 가까워지도록 스네이크의 불연속을 방지하기 위한 항과 정점들이 서로 동일한 거리를 유지한 채 스네이크의 급격한 꺾임을 방지하는 에너지항으로 구성된다. 여기서 $v_s$, $v_{ss}$는 s에 대한 1차 미분과 2차 미분을 나타낸다. 또한, 외부에너지는 영상의 명암 기울기 함수로써 밝기 값의 차이가 큰 부분에서 작은 값을 가지게 되어 스네이크는 밝기 값이 큰 곳(에지)을 찾아간다.

$$\begin{aligned} E_{int}(v(s)) &= \alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2 \\ E_{ext}(v(s)) &\approx -|\nabla \mathrm{I}(x,y)| \end{aligned} \tag{3}$$

식(2)에 정의된 총에너지를 전개하면 다음과 같다.

$$E^*_{snake}(v) = \int_0^1 (E_{ext}(v(s)) + \frac{1}{2}(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2))ds \quad (4)$$

$E^*_{snake}(v)$을 최소화 하는 $v(s)$를 찾기 위해서는 변분법을 이용한다. 스네이크에 작용하는 총에너지가 최소값을 가지기 위한 필요조건은 식(5)의 Euler-Lagrange 방정식을 만족해야 한다. 식(5)에서 $F_v$는 $v$에 대한 편미분, $F_{v_s}$는 $\frac{dv}{ds}$에 대한 편미분, $F_{v_{ss}}$는 $\frac{d^2v}{ds^2}$에 대한 편미분에 해당한다.

$$F_v - \frac{\partial}{\partial s}F_{v_s} + \frac{\partial^2}{\partial s^2}F_{v_{ss}} = 0 \quad (5)$$

식(5)에 따라 각 항을 전개하고 정리한 후, 초기 상태에 따라 Euler-Lagrange 방정식을 풀기 위해 반복적인 방법을 사용하여 식(6)과 같은 최종식을 얻을 수 있다(박찬모(1995); 김원(1998)). 여기서 A는 5단 대각행렬이며 fx(x, y)는 외부에너지의 x, y 방향의 편미분이고 $\Upsilon$은 step size이다.

$$\begin{aligned} x_t &= (\mathrm{A}+\gamma\mathrm{I})^{-1}(\gamma x_{t-1} - f_x(x_{t-1}, y_{t-1})) \\ y_t &= (\mathrm{A}+\gamma\mathrm{I})^{-1}(\gamma y_{t-1} - f_y(x_{t-1}, y_{t-1})) \end{aligned} \quad (6)$$

이러한 Kass의 스네이크는 초기 정점들이 윤곽선에서 멀리 설정되면 잘못된 결과를 얻을 수 있다는 단점과 윤곽선이 오목한 부분에서는 스네이크의 활동이 지체되는 문제점을 가지고 있다.

## 3. Gradient Vector Flow

Kass 스네이크의 두 가지 문제점을 해결하기 위해 Xu(1998)는 GVF를 제안하였다. Xu는 스네이크의 총에너지중 내부에너지항과 외부에너지항의 균등함으로 인하여 오목한 부분의 윤곽선 추출이 어렵다는 것을 확인하고 이진 영상이나 그레이 영상에 대한 에지맵을 식(7)과 같이 계산한다(Jain(1989)).

$$\text{edge map}\,(\nabla f): f(x,y) = -E^{(i)}_{ext}(x,y) \quad (7)$$

식(7)과 같은 에지맵은 <표1>과 같은 성질을 가지고 있으며 이런 성질을 개선하기 위하여, Xu는 영상에 대한 $\mathbf{V}(x,y) = (u(x,y), v(x,y))$의 벡터장을 만들고 식(8)과 같은 외부에너지 함수를 정의하여 최소화한다. 이것이 GVF장이다.

<표 1> 에지맵 성질

- 에지맵의 기울기 벡터($\nabla f$)는 에지쪽을 가리키며, 에지에 대해 직각이다.
- 기울기 벡터는 에지 근처에서 큰 크기를 가진다.
- 기울기 벡터는 영상의 동성(homogeneous) 부분에서 거의 0이다.

$$\epsilon = \iint \left(\mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |v - \nabla f|^2\right) dx dy \tag{8}$$

식(8)에서 GVF가 에지로부터 멀리 있을 때, 즉 $|\nabla f|$가 작을 때는 백터장의 편미분에 대한 제곱의 합으로 에지를 크게 하고, 반대로 GVF가 에지로부터 가까이 있을 때, 즉 $|\nabla f|$가 클 때는 $v \approx \nabla f$가 되어 에너지를 최소화 한다. 식(8)에 대한 Euler 방정식은 식(9)와 같으며, 여기서 $\nabla$는 Laplician operator이고, 식(9)는 벡터장의 Helmoholtz theorem(Gupta(1996))을 이용하여 구한다.

$$\begin{cases} \mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \\ \mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \end{cases} \tag{9}$$

이제 식(9)를 Kass 스네이크와 같이 반복적인 해를 구하기 위하여 시간(t)과 간소화 과정을 거치면 식(10)과 같다. 여기서 $b(x,y)$, $c^1(x,y)$와 $c^2(x,y)$는 반복과정 중에 고정된 값이다.

$$\begin{aligned} & u_t(x,y,t) = \mu \nabla^2 u(x,y,t) - b(x,y)u(x,y,t) + c^1(x,y) \\ & v_t(x,y,t) = \mu \nabla^2 v(x,y,t) - b(x,y)v(x,y,t) + c^2(x,y) \\ & \begin{cases} b(x,y) = f_x(x,y)^2 + f_y(x,y)^2 \\ c^1(x,y) = b(x,y) f_x(x,y) \\ c^2(x,y) = b(x,y) f_y(x,y) \end{cases} \end{aligned} \tag{10}$$

식(10)에서 기호의 혼동을 피하기 위하여, i←x, j←y, n←t, $\Delta$x,$\Delta$y←픽셀거리, $\Delta$t←time step로 정의하여 편미분을 구하면 식(11)과 같고, 이 값을 식(10)에 대입하여 정리하면 최종 GVF의 반복적인 해는 식(12)와 같이 구할 수 있다.

$$\begin{aligned} u_t &= \frac{1}{\Delta t}(u_{i,j}^{n+1} - u_{i,j}^n) \\ v_t &= \frac{1}{\Delta t}(v_{i,j}^{n+1} - v_{i,j}^n) \\ \nabla^2 u &= \frac{1}{\Delta x \Delta y}(u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j}) \\ \nabla^2 v &= \frac{1}{\Delta x \Delta y}(v_{i+1,j} + v_{i,j+1} + v_{i-1,j} + v_{i,j-1} - 4v_{i,j}) \end{aligned} \tag{11}$$

$$\begin{aligned} & u_{i,j}^{n+1} = (1 - b_{i,j}\Delta t)u_{i,j}^n + r(u_{i+1,j}^n + u_{i,j+1}^n + u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n) + c_{i,j}^1 \Delta t \\ & v_{i,j}^{n+1} = (1 - b_{i,j}\Delta t)v_{i,j}^n + r(v_{i+1,j}^n + v_{i,j+1}^n + v_{i-1,j}^n + v_{i,j-1}^n - 4v_{i,j}^n) + c_{i,j}^2 \Delta t \\ & r = \frac{\mu \Delta t}{\Delta x \Delta y} \end{aligned} \tag{12}$$

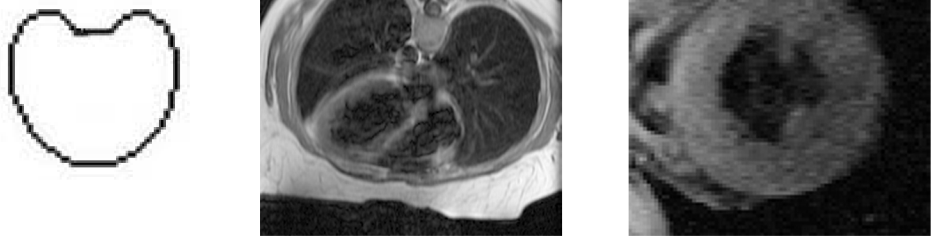
이런 GVF의 가장 큰 장점은 오목한 부분의 윤곽선 추출이 가능하고 스네이크의 초기 설정 범위가 기존 스네이크보다 넓다. 그러나 이미지의 에지맵과 벡터장을 계산하며, 편미분을 구하기 위하여 많은 계산시간이 필요하다. Xu는 256×256 픽셀 영상에 대한 실험에서 기존 스네이크 알고리즘은 8초, distance potential forces(Cohen(1993))는 155초, GVF는 420초가 걸렸다고 한다. 이런 계산시간은 2차원 영상보다는 3차원 영상에서 보다 더 많은 계산 시간을 필요로 할 것이다. 따라서 본 논문에서는 기존

스네이크 알고리즘과 GVF에 대한 계산시간을 실험을 통하여 비교 분석하고자 한다.

## 4. 실험 결과

Kass 스네이크와 GVF의 계산시간 비교는 <그림 1>과 같은 기초 영상 1개와 실제 영상 2개를 대상으로 실험하였다. <그림 1>의 (a)는 128×128의 jpg 파일이고 (b)는 512×512, (c)는 160×160 크기의 jpg 파일을 사용하였다. 논문의 목적이 스네이크의 정확한 성공률 보다는 계산시간의 비교를 측정하는 것이므로 <그림 1>의 (a)는 Xu의 그림을 수정해서 사용하였다.

또한 스네이크와 GVF의 구현은 Matlab을 사용하였는데, 논문에서 사용한 Matlab 소스 파일은 Xu의 원본 파일을 수정하여 사용하였다. 실험에서 스네이크와 GVF의 초기 정점은 사용자가 직접 지정하였고, 양 알고리즘 모두 동일하게 초기 정점을 설정하였다. 또한, 정확한 계산시간 측정을 위하여 stand-alone 방식으로 알고리즘을 실행하였다.

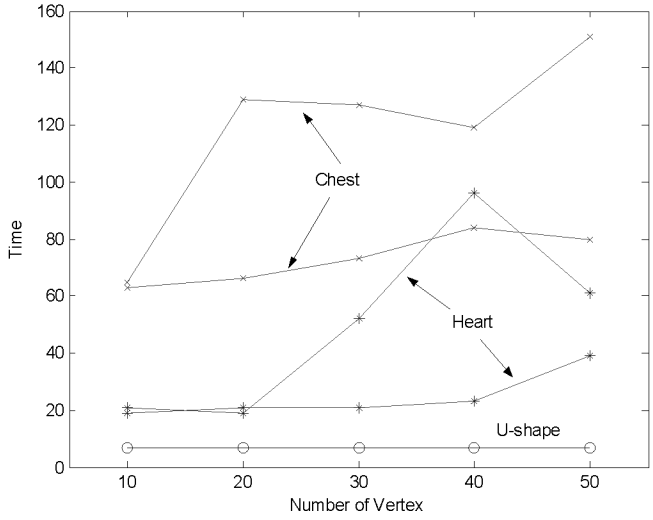


(a) U-Shape (b) Chest (c) Heart

<그림 1> 실험 영상

<표 2> 실험 영상의 계산시간

| | U-shape | | Chest | | Heart | |
|---|---|---|---|---|---|---|
| 정점의 수 | Snakes | GVF | Snakes | GVF | Snakes | GVF |
| 10 | 6.7 | 6.8 | 63 | 65 | 19 | 21 |
| 20 | 6.7 | 6.9 | 66 | 129 | 21 | 19 |
| 30 | 6.7 | 6.8 | 73 | 127 | 21 | 52 |
| 40 | 6.7 | 6.8 | 84 | 119 | 23 | 96 |
| 50 | 6.7 | 6.8 | 80 | 151 | 39 | 61 |
| 총 시간 | 33.5 | 34.1 | 366 | 591 | 123 | 249 |
| 평균 시간 | 6,7 | 6.8 | 73.2 | 118.2 | 24.6 | 49.8 |




<그림 2> 계산시간의 비교

<표 2>는 각 실험 영상에서 초기 정점 수의 변화에 따른 소요되는 계산시간의 결과이다. 여기서 단위는 초이며, Matlab의 cputime 함수를 이용하여 측정했다. 그리고 반복횟수는 U-shape=100, Chest, Heart=200까지 실시하였다.

우선 기초 영상인 U-shape 경우에는 이미지 자체가 단순하고 크기가 작기 때문에 스네이크와 GVF의 계산시간에는 큰 차이가 없다. 그러나 Chest의 경우 정점의 수가 증가함에 따라 스네이크와 GVF의 계산시간도 같이 증가한다. 특히, GVF의 계산시간이 평균적으로 스네이크보다 1.6배 더 소요됨을 알 수 있다. 그리고 Heart의 경우, U-shape와 이미지 크기의 차이가 크게 나지는 않지만 GVF의 계산 시간은 스네이크보다 약 2배 정도 더 소요되었다. <그림 2>는 <표2>의 결과를 그래프로 나타낸 것이다.

<그림 2>에서 Chest의 경우, 정점 10에서는 스네이크와 GVF의 계산시간이 비슷하다. 이때 초기 정점의 위치는 스네이크와 GVF기 동일하다. 그러나 정점을 20개로 늘리면서 초기 정점의 위치를 새로이 스네이크와 GVF에 설정하였다. 즉, 정점 10과 정점 20의 초기 위치는 다르다. 이와 같이 정점의 수에 따라 정점의 위치 설정이 다른 경우, <그림 2>에서와 계산시간이 선형적으로 증가하는 것이 아니라 각 초기 정점의 위치에 따라 계산시간이 불규칙적으로 나타남을 알 수 있었다. Heat의 경우에도 정점의 수가 40, 50에서 불규칙한 계산시간을 알 수 있다.

GVF에 대한 실험에서 에지맵의 계산은 한번만 수행하며 기초 영상에서는 2초, 실제 영상에서는 4초 정도 더 걸렸다. 이런 에지맵의 계산은 이미지의 크기에 따라 비례적으로 증가하는 계산량을 보였다. 그리고 계산시간 비교에서 Chest(40정점), Heart(40, 50정점)에서 처럼, 불규칙한 현상이 발생하는데 이것은 초기 정점의 위치에 따라 다른 결과를 나타내었고 앞으로 이에 대한 연구가 더 필요하다.

스테이크나 GVF의 응용에서 영상의 크기가 작은 경우는 GVF의 적용이 짧은 계산시간과 좋은 성공률을 나타낸다. 그러나 영상의 크기가 큰 경우는 영상의 복잡도에 따라, 단순한 영상인 경우는 짧은 계산시간이 걸리는 스테이크가 유리하며, 복잡한 영

상에 대한 좋은 성공률을 얻기 위해서는 계산시간이 많이 걸리더라도 GVF의 적용이 더 적합하다.

## 5. 결론

윤곽선 추출 알고리즘은 디지털 영상의 편집과 변형을 위하여 사용되는 기초적인 알고리즘이다. 이런 윤곽선의 동적인 추출 방법으로 초기에 스네이크가 개발되었고 그 후 GVF로 발전하였다. GVF는 스네이크의 단점인 초기 정점 설정 문제와 오목부분 윤곽선 추출을 해결할 수 있지만 에지맵과 편미분 계산에 상당한 계산시간이 소요된다. 따라서 본 논문에서는 스네이크와 GVF의 계산시간을 비교 분석하였다.

실험 결과, 기초 영상에서는 계산시간이 큰 차이가 없지만 실제 영상에서는 GVF가 기존 스네이크보다 약 2배 더 계산시간이 소요되었다. 그리고 계산시간도 정점의 수에 따라 선형적으로 증가하는 것이 아니라 초기 정점의 위치에 따라 다를 수 있다는 결과를 얻었다. 스네이크와 GVF의 응용에 있어서 영상의 크기가 작고 단순한 영상인 경우는 스네이크가, 복잡하고 높은 정밀도를 요구하는 응용에서는 GVF의 적용이 더 합리적인 선택이 된다.

## 참고 문헌


1. 곽영태(2006). An Initialization of Active Contour Models(Snakes) using Convex Hull Approximation, *한국데이터정보과학회지*, 17, 753-762.
2. 김원(1998). *이동물체 추적을 위한 이미지 플로우를 이용한 Active Contour 모델*, 한국과학기술원, 석사학위 논문.
3. 박찬모(1995). *Active Contour Models(Snakes)를 이용한 사용자의 손가락 추적에 관한 연구*, 포항공과대학, 석사학위 논문.
4. 장동혁(2003). *New Visual C++을 이용한 디지털 영상처리의 구현*, 와이어미디어.
5. Cohen, L. D. and Cohen, I. (1993). Finite-element methods for active contour models and balloons for 2-D and 3-D images, *IEEE Trans. on Pattern Anal. Machine Intell.*, 15, 1131-1147.
6. Gupta, S. N. and Prince, J. L. (1996). Stochastic models for DIV-CURL optical flow methods, *IEEE Signal Processing Letters*, 3, 32-35.
7. Jain, A. K. (1989). *Fundamentals of Digital Image Processing*, Prentice Hall, Engelwood Cliffs, NJ.
8. Kass, M., Witkin A., and Terzopoulos D. (1987). Snakes: Active contour models, *Int. J. Computer Vision,* 1, 321-331.
9. Petrou, M. and Bosdogianni P. (1999). *Image Processing : The Fundamentals*, John Wiley & Sons.
10. Xu, C. and Prince, J. L. (1998). Snake, Shapes, and Gradient Vector Flow, *IEEE Trans on Image Processing*, 7, 359-369.


[ 2007년 4월 접수, 2007년 5월 채택 ]