A SNAKE-BASED SEGMENTATION ALGORITHM FOR OBJECTS WITH BOUNDARY CONCAVITIES

Shin-Hyoung Kim, Ashraf Alattar and Jong Whan Jang

Department of Information and Communication Engineering, PaiChai University
Daejeon, The Republic of Korea
{zeros, ashraf and jangjw}@pcu.ac.kr

ABSTRACT

Concavities in the boundary of an object pose a challenge to active contour (snake) methods. 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, and is insensitive to the number of initial snake points. Experimental results have shown that our algorithm produces more accurate segmentation results than the conventional algorithm.

1. INTRODUCTION

Object segmentation is the basis for many important applications such as machine vision, video games, 3D interactive TV, interactive multimedia systems, and image and video coding [1]. In fact, the recently developed MPEG-4 standard [2] is an object-based image and video coding algorithm that requires object segmentation in the first stage.

Over the past two decades, various object segmentation schemes have been developed for extracting an object from an ordinary image. The most recent of these algorithms is the active contour (snake) algorithm [3]-[6]. A snake is an energy-minimizing spline guided by internal forces that preserve its characteristics, and external forces that pull it toward image features such as lines and edges. The snake approach suffers from some challenges such as the difficulty to progress into boundary concavities. Xu *et al.* proposed the GVF (gradient vector flow) snake [5] to handle the concavities problem. The GVF method uses a spatial diffusion of the gradient of the edge map of the image instead of using the edge map directly as an external force. Although the method has the advantages of insensitivity to initialization and the ability to move into boundary concavities, it can not handle gourd-shaped concavities because of the concentration of GVF energy in the neck of the gourd.

In this paper we present a snake-based method for object segmentation addressing the above challenges through a modified snake model with optimized points. We propose new modifications to the energy formulation of the snake model to handle the concavity problem. Optimization of snake points is addressed by managing the insertion of new points and deletion of unnecessary points to better describe the object's boundary. Experiments demonstarted the insensitivity of the proposed algorithm to the number of initial snake points.

This paper is organized as follows. Section 2 covers a background on the snake model. Our proposed method for extracting objects with boundary concavities using an optimized snake is presented in Section 3. In Section 4 we show results of the simulation for evaluating the performance of our method, and the conclusions are given in Section 5.

2. THE SNAKE MODEL

The snake algorithm was first introduced by Kass *et al.* [3]. A snake is the energy-minimizing spline guided by internal constraint forces and influenced by external image forces that pull it toward features such as lines and edges.

In the discrete formulation of a snake, the contour is represented as a set of snake points $v_i = (x_i, y_i)$ for $i = 0,...,M-1$ where x_i and y_i are the x and y coordinates of a snake point, respectively and M is the total number of snake points. The snake model is based on the definition of an energy function which is typically written as follows:

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

The function contains two energy components. The first component is called the *internal energy* and it concerns contour properties such as curvature and discontinuity. The second component is the *external energy* which is typically defined by the gradient of the image at a snake point.

3. PROPOSED ALGORITHM

Our proposed algorithm modifies the conventional snake model, and introduces a new mechanism for the insertion and/or deletion of snake points as necessary.

3.1 Proposed new snake energy function

In this section, we describe our special snake energy function defined for a 2D image. Internal energy is typically expressed in terms of two terms: *continuity energy*, and *curvature energy*. Our modifications to these two terms and to the external energy term will be explained in the following subsections

3.1.1 Internal energy terms

The first internal energy term is the continuity energy term. The main role of the conventional continuity energy term is to make even spacing between the snake points by minimizing the difference between the average distance and the distance between neighboring snake points. In this arrangement point spacing is globally and uniformly constrained by the average distance. This is not suitable for objects with concavities because the distance between snake points should be relaxed in boundary concavities. Therefore, in our method, point spacing in boundary concavities is allowed to differ from other parts of the snake. We define the normalized continuity energy as follows:

$$E_{con}(v_i) = \frac{\left\| \|v_{i-1} - v_i\| - \|v_i - v_{i+1}\| \right\|}{con_{\max}} \tag{2}$$

where $con_{\max}$ is the maximum value in the search neighborhood. The proposed continuity energy can make even spacing between neighboring snake points in concave parts of the object's boundary.

For curvature energy, the goal is to control the smoothness of the contour between neighboring snake points. The way in which this term is formulated affects the ability of snake points to progress into boundary concavities, and conventional snake algorithms show poor performance in this aspect. In this paper we present a new curvature energy solving the problem based on the Frenet formula. As depicted in Fig. 1(a), let $\Psi : I \rightarrow \Re^3$ be a unit-speed curve. Then, $T = \Psi'$ is the unit tangent vector field on Ψ, and $N = T' / \|T'\|$ is the principal normal vector field of Ψ; where $\|T'\|$ is the length of the curvature. The vector field $B = T \times N$ on Ψ is the binormal vector field of Ψ, and the sign of the z-dimension of B is positive if B is upward and is negative if it is downward. Therefore, we consider the sign of the binormal vector. In 2D the sign of the binormal vector $B(v_i)$ can be obtained using the cross product of the two vectors $T(v_i)$ and $N(v_i)$ as follows:

$$B(v_i) = T(v_i) \times N(v_i) = \begin{vmatrix} x_i^T & y_i^T \\ x_i^N & y_i^N \end{vmatrix} = (x_i^T y_i^N - y_i^T x_i^N)\vec{e}_z \tag{3}$$

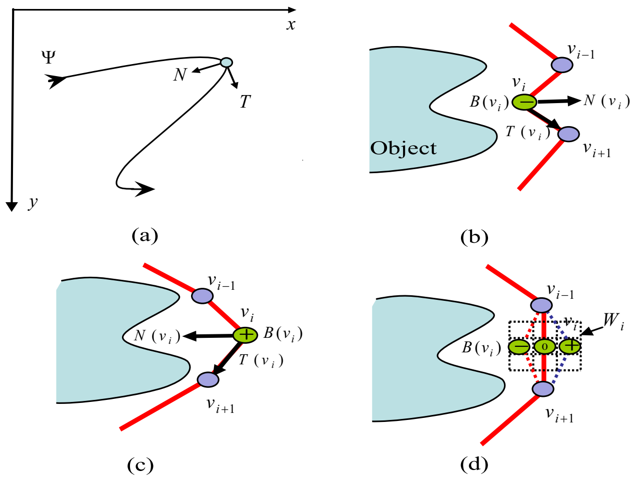

Fig. 1. Movement of snake points based on the binormal vector.

where (x_i^T , y_i^T) and (x_i^N , y_i^N) are the x and y coordinates of the tangent vector $T(v_i)$ and normal vector $N(v_i)$ at the current point v_i , respectively. $\vec{e}_z$ is a unit vector of the z-component. In the discrete formulation of a snake point, $T(v_i)$ and $N(v_i)$ are defined as follows:

$$T(v_i) = v_{i+1} - v_i \tag{4}$$

$$N(v_i) = v_{i-1} - 2v_i + v_{i+1} \tag{5}$$

In the case that a snake point is outside the object (as in Fig. 1(b, c, d)), the movement of a snake point can be described as follows:

i) When $B(v_i)$ is negative, as the case in Fig. 1(b), v_i moves in the opposite direction of $N(v_i)$ seeking a location with maximum value of $\|N(v_i)\|$.

ii) When $B(v_i)$ is positive, as the case in Fig. 1(c), the snake point v_i must take the direction of $N(v_i)$ to minimize curvature and to converge on the boundary of the object.

iii) In the case of Fig. 1(d), the value of $\|N(v_i)\|$ at v_i is zero because the point is on a straight line, and, therefore, $B(v_i)$ has a zero vector at v_i. In this case, we examine neighboring points located within the window W_i . A candidate point with a negative $B(v_i)$, such as the point to the left of v_i , maximizes the $\|N(v_i)\|$ and thus v_i must move to that location.

The normalized curvature energy is defined as Eq. (6) [6]. It is further weighted by the constant λ , as given in Eq. (7).

$$E_c(v_i) = \left\| \, T(v_i)/\|T(v_i)\| - T(v_{i-1})/\|T(v_{i-1})\| \, \right\| \quad (6)$$

$$E_{cur}(v_i) = \lambda E_c(v_i) \quad (7)$$

The constant λ can be set for different applications, relative to the sign of $B(v_i)$ on the contour. In our work, λ is set to +1 when the sign of $B(v_i)$ is positive and −1 otherwise.

3.1.2 External energy function

The external energy is defined as $-|\nabla[G_\sigma(v_i) * f(v_i)]|/e_{\max}$, where $G_\sigma(v_i)$ is a two-dimensional Gaussian function with standard deviation σ and ∇ is the gradient operator. $f(v_i)$ is the image intensity function, and $e_{\max}$ is the maximum value in the search neighborhood.

3.1.3 Total snake energy

The proposed snake energy function is defined as follows:

$$E_{snake}(v) = \sum_{i=0}^{M-1}\left(\alpha E_{con}(v_i) + \beta E_{cur}(v_i) + \gamma E_{ext}(v_i)\right) \quad (8)$$

The parameters α , β and γ are set to 1.0 ,1.0 and 1.2 respectively.

3.2 Optimizing the number of snake points

After snake points converge in the first step to the boundary of the object, additional points are inserted and unnecessary points are deleted to better describe the boundary concavities. This step is explained as follows.

3.2.1 Point insertion and deletion

The decision to insert a new point between two points v_i and v_{i+1} depends on the length of the normal vector at v_i , $\|N(v_i)\|$. If $\|N(v_i)\|$ is above a threshold th_N , an additional point is inserted. This condition can be expressed by the Boolean expression:

$$\|N(v_i)\| \geq th_N \quad (9)$$

and the newly inserted point is defined as $c_i = (v_i + v_{i+1})/2$.

On the other hand, when the $\|N(v_i)\|$ at a point v_i is small enough to fall below th_N , the point is likely to be on a straight line. In such case, v_i can be safely removed because the boundary can be described well enough by the two snake points adjacent on each side.

3.2.2 Movement of inserted points

Fig. 2 illustrates the decision controlling the movement of inserted points taking into consideration whether the inserted point is inside or outside the object.

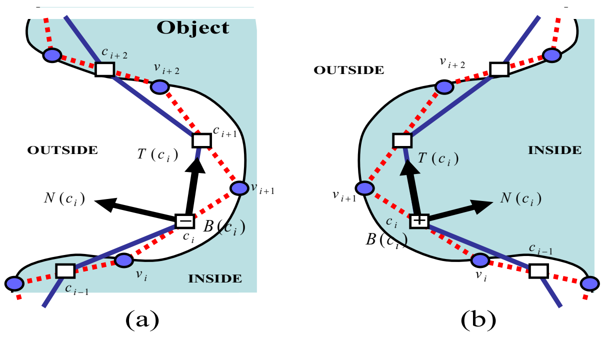

Fig. 2. Movement of inserted points in convex/concave segments of the object's boundary

As shown in the Fig. 2, the sign of $B(c_i)$ is negative in concave segments and positive in convex segments. To decide the movement of an inserted point, we first check the sign of its binormal vector $B(c_i)$ relative to previous and next inserted points. If $B(c_i)$ is negative, then the point is outside, and in that case we apply exactly the same scheme explained in section 3.1.1 above. On the other hand, if the inserted point turns out to be inside (i.e., $B(c_i)$ is positive), we apply the same scheme except that we reverse the signs for λ .

4. EXPERIMENTAL RESULTS

To verify the performance of our algorithm a set of experiments using computer simulation has been performed. The algorithm was coded in Visual C++ 6.0, and the simulation was performed on a Pentium IV machine with 1GByte of memory running at 3GHz clock speed. We used binary and color real images with 320×240 image size.

The criteria used to verify the accuracy of the estimated snake region R_{est}, compared with the original object region R_{ori} is *Relative Shape Distortion*, *RSD*(*R*) , which is defined as $RSD(R) = \left(\sum_{(x,y)\in f} R_{ori}(x,y) \oplus R_{est}(x,y)\right) / \sum_{(x,y)\in f} R_{ori}(x,y)$, where the $\oplus$ sign is the binary XOR operation. The simulation process and its results are illustrated in Figs. 3 and 4.

Fig. 3 shows results of an experiment on a binary image for an object with a gourd-shaped concavity. The greedy snake could not detect the boundary of the object (a), and the GVF snake points failed to proceed beyond the neck of the gourd (b). With our proposed algorithm the snake points converged better onto the boundary concavity in only three iterations of insertion (c). Fig. 3(d) shows the results for our proposed algorithm on a real image. The snake points converged onto the gourd-shaped concavities in four iterations.

Fig. 4 shows the relationship between the number of initial snake points and the number of iterations.

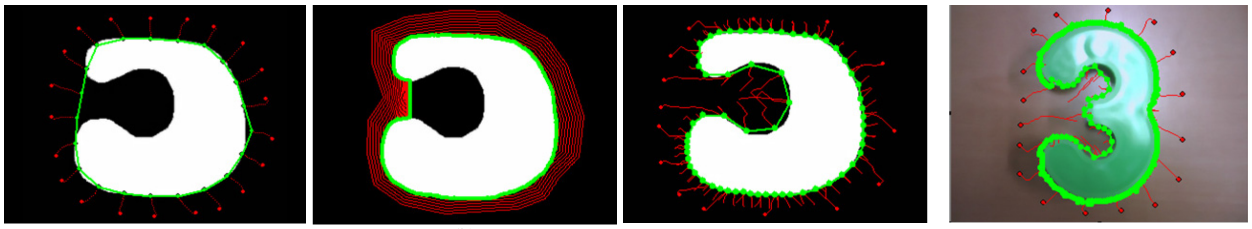

Fig. 3. Results of experiment on the gourd-shaped object: (a) Greedy snake, (b) GVF snake, (c)-(d) Proposed algorithm

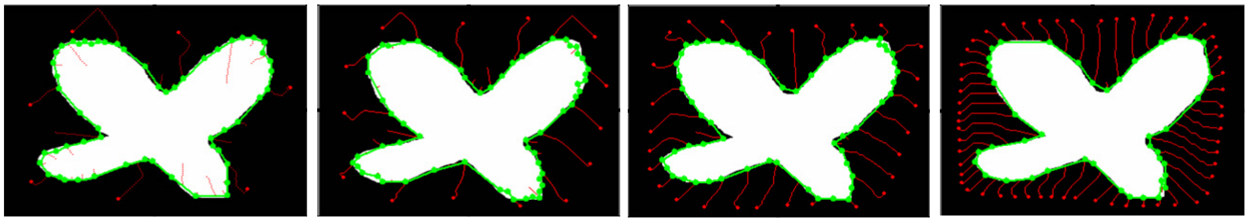

Fig. 4. Results of experiment on the relations between the number of initial snake points and the number of iterations

When the snake is initialized with sufficient number of points, convergence is achieved with less iterations and point insertion. However, a snake that is initialized with a few number of points, requires an increased number of iterations with point insertions. The conventional methods in [3] and [4] can not optimize the number of sanke points, and thus do not handle boundary concavities which require more snake points. Table 1 summarizes the results shown in Fig. 4, and gives performance measures in terms of *RSD*.

Table 1. Performance comparison in terms of *RSD*

	Number of initial snake points	Number of iteration (insertion & deletion)	Final number of snake points	*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

5. CONCLUSIONS

In this paper, we have presented a new snake-based segmentation algorithm for objects with boundary concavities. Our algorithm extends and improves the conventional snake model. The developed algorithm was tested and showed successful results in extracting objects with boundary concavities.

In the proposed new snake algorithm, the movement of snake points is determined using the sign of the cross product of the tangent and normal vectors. In addition, we proposed optimizing the number of snake points to better describe the object's boundary. In consequence, we can solve the problem which occurs with gourd-shaped boundary concavities. Performance has been evaluated by running a set of computer simulations using 2D image data having objects with varying degrees of boundary concavity. When compared with conventional snake algorithms, our method has shown a superior object segmentation capability in terms of accuracy. Further research work is being considered to follow up from object segmentation to object tracking in video sequences.

6. REFERENCES

[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. Sandvancd, 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 Tech nology - Coding of Audio-Visual Objects Part2: Visual " *ISO/IEC 14496-2,* July 2001.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snake: Activ e 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 Ac tive Contours And Curvature Estimation," *Computer Vi sion, 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. Image Processing,* vol. 7, No. 3, pp. 359-369, March 1998.

[6] S. H. Kim and J. W. Jang, "Snakes For Object Contour Tracking Stereo Sequences," *ICME 2005*, pp. 674-677, July 2005.