Segmentation of confocal microscopic image of insect brain

Ming-Jin Wu[a], Chih-Yang Lin[a], Yu-Tai Ching[a]
[a]Department of Computer and Information Science, National Chiao-Tung University, Taiwan

ABSTRACT

Accurate analysis of insect brain structures in digital confocal microscopic images is valuable and important to biology research needs. The first step is to segment meaningful structures from images. Active contour model, known as snakes, is widely used for segmentation of medical images. A new class of active contour model called gradient vector flow snake has been introduced in 1998 to overcome some critical problems encountered in the traditional snake. In this paper, we use gradient vector flow snake to segment the mushroom body and the central body from the confocal microscopic insect brain images. First, an edge map is created from images by some edge filters. Second, a gradient vector flow field is calculated from the edge map using a computational diffusion process. Finally, a traditional snake deformation process starts until it reaches a stable configuration. User interface is also provided here, allowing users to edit the snake during deformation process, if desired. Using the gradient vector flow snake as the main segmentation method and assist with user interface, we can properly segment the confocal microscopic insect brain image for most of the cases. The identified mushroom and central body can then be used as the preliminary results toward a 3-D reconstruction process for further biology researches.

Keywords: **Keywords:** active contour, snake, gradient vector flow, segmentation, confocal microscopic image

1. INTRODUCTION

Construction of a standard brain plays an important role in the analysis of brain's anatomy and functions. The first step is to segment structures in the brain image. Successful analysis highly depends on the validity of the segmentation. Segmentation is one of the most difficult tasks in image processing. The challenge is to extract boundary elements that belong to the same structure and integrates these elements into a coherent and consistent model of structure. Traditional low-level techniques such as thresholding and edge detecting which only utilize local information can make mistakes during integration step due to lack of global geometry properties of the object.

Alternatively, the active contour model, known as snakes, which has been introduced by Kass, Witkin and Terzopoulos in 1987 [1], integrates the image feature extraction and representation phase into a single process. A snake is an energy-minimizing curve defined within an image domain that moves under the influences of internal forces coming from within the curve itself and external forces computed from image data. The internal and external forces are defined such that the snake will be attracted to an object boundary or other features within an image. Snakes are widely used in many applications, including edge detection [1], shape modeling [2], segmentation [3], and motion tracking [4]. We will adopt active contour model to segment the mushroom body and the central body in insect brain image.

2. ACTIVE CONTOUR

A traditional active contour model is a mapping: $\Omega = [0, 1] \rightarrow R^2$, and $S \rightarrow v(s) = (x(s), y(s))$. Here we represent the position of a snake parametrically by $v(s) = (x(s), y(s))$. We define the model as a space of admissible deformation A and a functional E to be minimized. This functional represents the energy of the model and has the form

$$E : A \rightarrow R,$$

$$v \rightarrow E(v) \int_0^1 E_{int}(v(s)) + E_{ext}(v(s)) ds, \quad (1)$$

where E_{int} represents the internal energy of the snake due to bending, and E_{ext} serves as external force and is derived from image features. E_{int} serves to impose smoothness constraint on the snake. E_{ext} pushes or pulls the snake toward desired features such as edges. As internal and external energy are formulated, we deform the snake by minimizing (1). The internal energy can be written as:

Medical Imaging 2002: Image Processing, Milan Sonka, J. Michael Fitzpatrick,
Editors, Proceedings of SPIE Vol. 4684 (2002) © 2002 SPIE · 1605-7422/02/$15.00

$$E_{\text{int}} = (\alpha(s)\left|v'(s)\right|^2 + \beta(s)\left|v''(s)\right|^2)/2 \tag{2}$$

where $\alpha(s)$ and $\beta(s)$ are weighting parameters that control the snake's elasticity and rigidity, $v'(s)$ and $v''(s)$ denote the first and second derivatives of the curve v with respect to s. The first-order term $\alpha(s)|v'(s)|^2$ makes the snake behave like a string, whereas the second-order term $\beta(s)|v''(s)|^2$ makes it behave like a rod. Position or tangent discontinuities may be introduced along a snake by setting $\alpha(s)$ or $\beta(s)$ to zero [12]. In order to make snakes useful, we need energy functions that attract snakes to salient features in images as long as it takes on its smaller values at these desired features. External energy can be derived from image, or be user defined. It can also be a weighted combination of two or more energy functions. If v in (1) is a local minimum for E, it satisfies the associated Euler-Lagrange equation:

$$(\alpha(s)v'(s))' - (\beta(s)v''(s))'' - \nabla E_{ext}(v(s)) = 0, \tag{3}$$

given $v(0)$, $v'(0)$, $v(1)$, $v'(1)$ as boundary conditions.

Traditional active contour model suffers from the following three key difficulties: First,. the evolution of a snake highly depends on its initial state. That means the active contour model has narrow capture range so that the initial contour must be close to the true boundary or else it may converge to the wrong boundary. Second, active contour model lacks the ability to handle boundary concavities. Third, snake has the tendency to become stagnant at a nearby local energy minimum. For the first and second problem, Xu and Prince proposed a method called gradient vector flow in 1998 [5] to solve the problems. For the third problem, the main difficulty is due to noise. Preprocessing of the image can be a great help. However, if the noise is too strong, it still requires users to manually adjust the snake to get rid of the local minimum.

3. METHOD

3.1 Gradient Vector Flow Snake

Xu and Prince proposed gradient vector flow (GVF) [5] to solve the first two problems encountered in the traditional active contour model. The method they proposed is a new class of external forces for active contour, a dense vector fields derived from images by minimizing certain energy functional. The minimization is achieved by solving a pair of decoupled linear partial differential equations that diffuse the gradient vectors of a gray-level or binary edge map computed from image. We will call an active contour that uses gradient vector flow as its external force as a GVF snake. A GVF snake has wide capture range, can progress into concavities, and is insensitive to initialization. Its initialization can be inside, outside, or across an object's boundary. By the diffusion process, capture range is increased.

First, we define an edge map $f(x,y)$ derived from the image $I(x,y)$ having the property that it is larger near the desired boundary. We can use any edge-detecting method or any gradient filter. Then, we will extend the gradient map farther away from edges and into homogeneous regions using a computational diffusion process. As an important benefit, the inherent competition of the diffusion process will also create vectors that point into boundary concavities.

We define the gradient vector flow field to be the vector field $p(x,y) = [g(x,y),h(x,y)]$ that minimize the energy functional

$$\xi = \iint \mu(g_x^2 + g_y^2 + h_x^2 + h_y^2) + \left|\nabla f\right|^2 \left|p - \nabla f\right|^2 dxdy\,. \tag{4}$$

When $|\nabla f|$ is small, the energy is dominated by sum of the squares of the partial derivatives of the vector field, yielding a slowly varying field. On the other hand, when $|\nabla f|$ is large, the second term dominates the integrand, and is minimized by setting $p = \nabla f$. This produces the desired effect of keeping p nearly equal to the gradient of the edge map when it is large, but forcing the field to be slowly-varying in homogeneous regions. The parameter μ is a regularization parameter governing the tradeoff between the first term and the second term in the integrand. This parameter should be set according to the amount of noise present in the image. It has been shown that the first term in

the integrand corresponds to an equal penalty on the divergence and curl of the vector field [6]. Therefore, the vector field resulting from this minimization can be expected to be neither entirely curl-free nor entirely divergent-free.

Using calculus of variations [7], it can be shown that the GVF field can be found by solving the following Euler equations:

$$\mu\nabla^2 g - (g - f_x)(f_x^2 + f_y^2) = 0\,, \tag{5a}$$

$$\mu\nabla^2 g - (g - f_x)(f_x^2 + f_y^2) = 0\,, \tag{5b}$$

where $\bigtriangledown^2$ is the Laplacian operator.

3.2 Reparameterization

When a snake deforms, the distances between vertices of the snake will change. This may result in local variation as well as in global changes in the resolution of the model. Both are undesirable, but slight local variation in the resolution is unavoidable. What we can do is to keep this variation between certain limits by periodically resample the snake along its path. We resample the snake according to a user defined parameter l_{des}. Then we resample the snake to keep the distance between the neighboring vertices between $0.5l_{\text{des}}$ and $1.5l_{\text{des}}$. When the distance is less than $0.5l_{\text{des}}$, we merge tow adjacent vertices into a single vertex. If the distance is longer than $1.5l_{\text{des}}$, we insert a vertex between the vertices [8].

3.3 Numerical Solution

We substitute (2) into (1) to get the following equation:

$$v \rightarrow E(v) = \int_0^1 \frac{1}{2}[\alpha(s)|v'(s)|^2 + \beta(s)|v''(s)|^2] + E_{ext}(v(s))ds\,. \tag{6}$$

We will take $\alpha(s)$ and $\beta(s)$ as a user-defined constant α and β, and is remained unchanged during the deformation process. A snake that minimizes E must satisfy the Euler equation:

$$\alpha v''(s) - \beta v''''(s) - \nabla E_{ext} = 0\,. \tag{7}$$

To find a solution to (7), the snake is made dynamic by treating v as function of time t as well as s. The partial derivative of v with respect to t is then set equal to the left hand side of (7) as follows:

$$v_t(s,t) = \alpha v''(s,t) - \beta v''''(s,t) - \nabla E_{ext}\,. \tag{8}$$

The equation becomes after finite differences in space (step h) and in time (step τ):

$$(I_d + \tau A)\mathbf{V}^t = (\mathbf{V}^{t-1} + \tau F(\mathbf{V}^{t-1}), \tag{9}$$

where A is a pentadiagonal matrix, I_d denotes the identity matrix, $\mathbf{V}$ and F denote the vectors of position v_i and forces $-\bigtriangledown E_{ext}(v_i)$. Thus, we obtain a linear system and we can easily solve the pentadiagonal banded symmetric positive system by LU decomposition. To solve (4), it can be shown by using the calculus of variations [7] that the GVF field in (4) can be found by solving the following Euler equations:

$$\mu\nabla^2 g - (g - f_x)(f_x^2 + f_y^2) = 0\,, \tag{10a}$$

$$\mu\nabla^2 h - (h - f_x)(f_x^2 + f_y^2) = 0\,. \tag{10b}$$

Equation (10a) and (10b) can be solved by treating *g* and *h* as functions of time and solving:

$$g_t(x,y,t) = \mu\nabla^2 g(x,y,t) - [g(x,y,t) - f_x(x,y)]\cdot[f_x(x,y)^2 + f_y(x,y)^2], \tag{11a}$$

$$h_t(x,y,t) = \mu\nabla^2 h(x,y,t) - [h(x,y,t) - f_x(x,y)]\cdot[f_x(x,y)^2 + f_y(x,y)^2]. \tag{11b}$$

The equation in (11) are known as generalized diffusion equations, and are known to arise in the fields of heat conduction, reactor physics and fluid flow [11]. We rewrite (11) as follows:

$$g_t(x,y,t) = \mu\nabla^2 g(x,y,t) - b(x,y)g(x,y) + c^1(x,y), \tag{12a}$$

$$h_t(x,y,t) = \mu\nabla^2 h(x,y,t) - b(x,y)h(x,y) + c^2(x,y), \tag{12b}$$

where $b(x,y) = f_x(x,y)^2 + f_y(x,y)^2$, $c^1(x,y) = b(x,y)\,f_x(x,y)$, and $c^2(x,y) = b(x,y)\,f_y(x,y)$.

To set up the iterative solution, let the indices *i,j* and *n* correspond to *x,y* and *t*, and let the spacing between pixels be Δx and Δy and the time step for each iteration be Δt. Then (12) can be approximated by finite differences as follows:

$$g_t = \frac{1}{\Delta t}(g_{i,j}^{n+1} - g_{i,j}^{n}), \quad h_t = \frac{1}{\Delta t}(h_{i,j}^{n+1} - h_{i,j}^{n}).$$

$$\nabla^2 g = \frac{1}{\Delta x \Delta y}(g_{i+1,j} + g_{i,j+1} + g_{i-1,j} + g_{i,j-1} - 4g_{i,j}),$$

$$\nabla^2 h = \frac{1}{\Delta x \Delta y}(h_{i+1,j} + h_{i,j+1} + h_{i-1,j} + h_{i,j-1} - 4h_{i,j}). \tag{13}$$

Substituting these approximations into (12) gives the iterative solution to GVF as follows:

$$g_{i,j}^{n+1} = (1 - b_{i,j}\Delta t)g_{i,j}^{n} + r(g_{i+1,j}^{n} + g_{i,j+1}^{n} + g_{i-1,j}^{n} + g_{i,j-1}^{n} - 4g_{i,j}^{n}) + c_{i,j}^{1}\Delta t, \tag{14a}$$

$$h_{i,j}^{n+1} = (1 - b_{i,j}\Delta t)h_{i,j}^{n} + r(h_{i+1,j}^{n} + h_{i,j+1}^{n} + h_{i-1,j}^{n} + h_{i,j-1}^{n} - 4h_{i,j}^{n}) + c_{i,j}^{2}\Delta t, \tag{14b}$$

where $r = \frac{\mu\Delta t}{\Delta x \Delta y}$. Convergence of the above iterative process is guaranteed by a standard result in the theory of numerical methods [12]. Equation (14) is stable whenever the restriction $r \le \frac{1}{4}$ is maintained.

3.4 Stopping Criterion

When is a snake converged? One usually uses as many iterations as necessary until the snake stops moving. This is equivalent to have the snake reach its lowest possible total energy E, given an initial configuration v(s,0) in the external force field. This means when the snake reaches a steady state (i.e., v_t=0), the snake might traverse along the desired valley of the external force field. But if the bottom of the valley is not at the same height, some point in the desired valley might still moving to seek for the nearby local minimum. In such cases, vertices of the snake will have a strong tendency to pile up in the deepest pocket of the valley. Gradient vector flow is especially vulnerable to this situation since it is a neither entirely divergent-free nor entirely curl-free vector field. Thus using the steady-state criterion is

inadequate as a stopping criterion. To solve this problem, we will use global snake geometry properties to define the stopping criterion. We will define the stopping criterion as the contour similarity. If the similarity between two snakes is within a threshold, we consider these two snakes are the same snake and confirm that the snake is converged. We will adopt a modified chain code curve encoding method [9] and longest common subsequence techniques to evaluate the similarity of curves. Chain code is first proposed by Freeman. The concept is to divide a curve into fixed length segment and encode the curve according the direction of segments along the curve. In general, encoding method has 4-connectivity (Fig. 1) and 8-connectivity (Fig. 2). We subdivide the plane into 4 (8 respectively) regions in 4-connectivity (8-connectivity respectively) for encoding. Given a start vertex on a curve, we compute the vector of adjacent vertices along the curve. Then we encode the curve into a bit string according to each vector's direction. We will call the bit in a bit string as *direction bit*. Figure 3 is an example in 8-connectivity encoding.

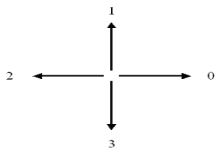


Figure 1. 4-connectivity

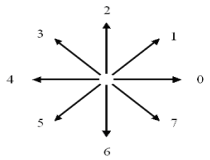


Figure 2. 8-connectivity

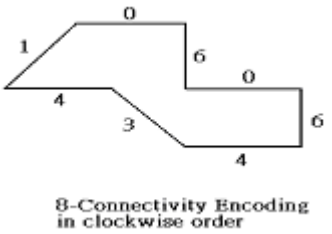


Figure 3. An chain code encoding example

When two curves are encoded into two bit strings, we find the longest common subsequence (LCS) of these strings by dynamic programming technique [10]. Suppose that LCS should be larger in two similar curves than that in tow less similar curves. We define a similarity function as follows:

$$S(v1, v2) = (Len(v1) + Len(v2) - 2 \times LCS(v1, v2)) / (Len(v1) + Len(v2)) \quad (14)$$

where S denotes the similarity function, $v1$ and $v2$ are curves to be matched, *Len* returns the length of the curve, *LCS* returns the length of the longest common subsequence of $v1$ and $v2$. Note that $0 \le S(v1, v2) \le 1$. If two curves are identical, $S(v1,v2) = 0$. More smaller the S is, more similar are the curves. Given a threshold T_s, if S is smaller than T_s, we consider that the snake has been converged. However, chain code has the following problem. In figure 4, vector **A** and vector **B** is in different encoding region. Thus vector **A** and **B** is considered pointing in different direction and are encoded into distinct direction bits. But if the angle in vector **A** and **B** is small, they should be considered pointing in the same direction. We made a little modification in the longest common subsequence algorithm to solve this problem. In the original longest common subsequence algorithm, a string bit represents vector's direction. Now we let each string bit represents vector itself, not only direction. The comparison of string bits is replaced by inner product of the vectors that to be compared. If the value of the inner product is larger than a given threshold value θ, we consider these two string bits are identical. After the modification, we provide a reliable stopping criterion.

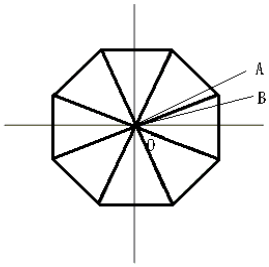


Figure 4.

3.5 User Interface

Snakes are easily trap in local minimum or converged to wrong boundary due to noise or ill initialization. When this happens, we need user assist to help the snake get rid of bad configuration. The other situation that needs user assist is topology maintenance. When an object splits or two objects merge, GVF snake can't handle topology change. We need user manually add or delete snakes if necessary.

So we provide the following user-interface in the segmentation process:

1. Add a new snake in the image
2. Delete an existing snake in the image
3. Replace a segment of curve of an existing snake with a new one in order to get rid of local minimum.
4. Change α, β (internal force coefficients), and l_{des} (reparameterization coefficient) of a certain snake during snake deformation process.

4. EXPERIMENTAL RESULT

We implement the GVF Snake under windows environment on personal PC equipped with AMD K6-2 400MHz CPU and 128Mb RAM. The confocal microscopic insect brain image data is 512x512 pixel, 256 in gray-level image, total 100 slices. Our goal is to semi-automatically segment the mushroom body and the central body in the image. Figure 5 indicates the region of interest.

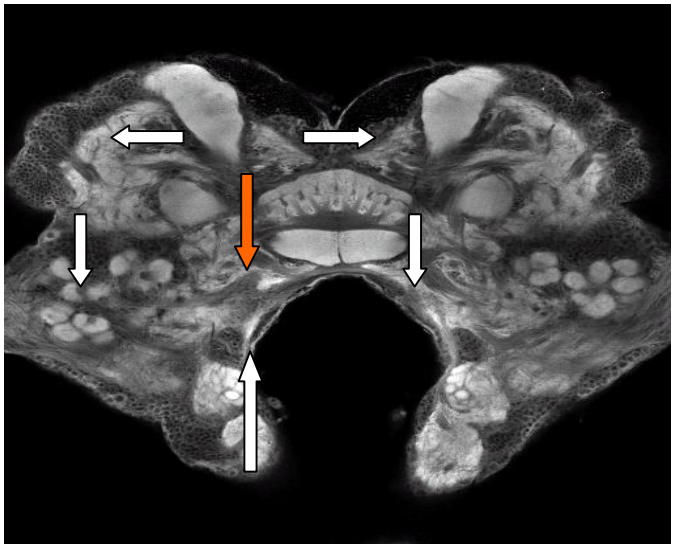
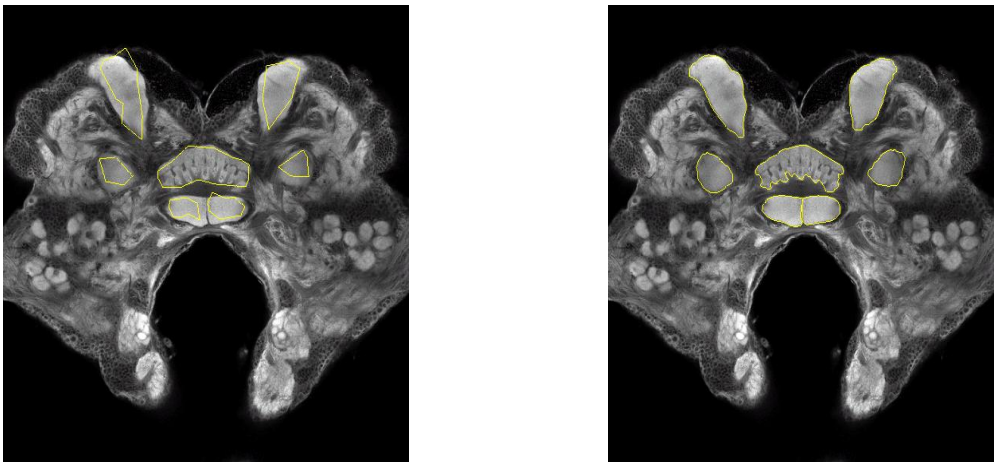


Figure 5. Red arrow points to the central body and white arrow points to mushroom body

We use Canny edge filter to calculate edge maps, μ=0.1, Δt=0.18 for all GVF and l_{des}=1 for all snakes. Each GVF is computed 512 iterations. For an N$\times$N pixel image, compute GVF for N iterations is reasonable since it is sufficient for an edge map to diffuse through the whole image. The average time complexity to calculate GVF for one slice is about 210 seconds.

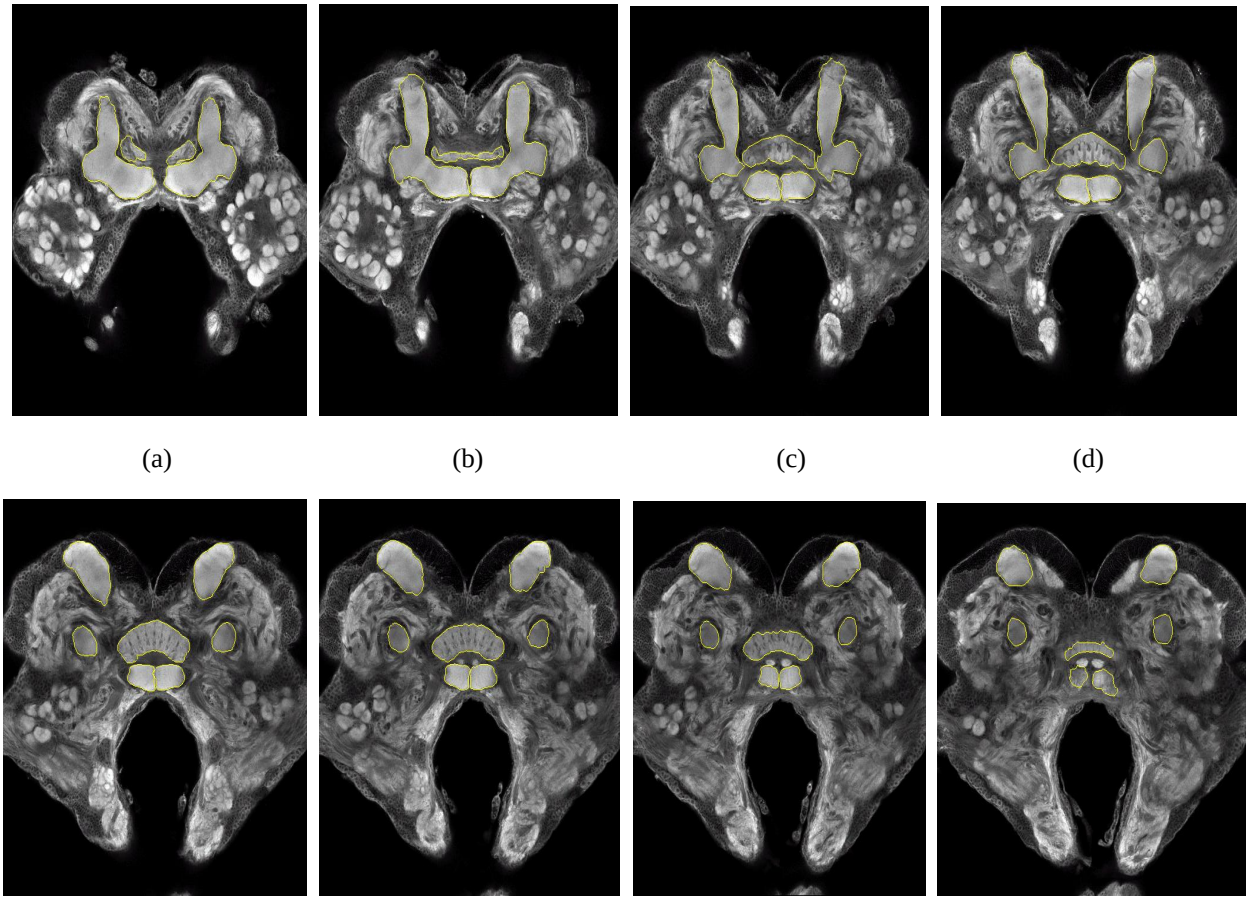
We will choose one slice as the first slice, and manually draw initial contours on the image. The image in figure 5 is taken as the first slice to be segmented. Figure 6 shows seven initial contours given in the image. While the first slice has been segmented, we save the final contours. These contours will be used as the initial contours in the adjacent slices.



(a) (b)

Figure 6.(a) Initial stateFigure, and (b) the results of (a)

Figure 7 show 8 slices of image segmented by using GVF snake and provided user interface assist. Some slices are well segmented. But for some image that is fuzzy in the region of interest, the result is not good due to noise and ill preprocessing of image.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7. Eight slices of image segmented by using GVF snake.

5. DISCUSSION

Active contour model as well as GVF snake highly depends on the external force field. In GVF snake, external force field is computed from edge map. If the original image is not well preprocessed, we get an ill edge map and the GVF snake may not well converge to the desired boundary.

Note that the central body in figure 8 and the mushroom body in figure 15 are not satisfactorily locates on the boundary because the image itself is fuzzy on these regions. Good preprocessing of image is necessary for better segmentation. For the confocal microscopic insect brain image, further texture analysis seems to be a feasible way for preprocessing.

In summary, we reviewed the active contour model and discuss its main drawbacks and introduce gradient vector flow snake that greatly reduces these drawbacks. A new converge criterion is also proposed. By demonstrating some examples in synthetic and clinical images, we showed the power of GVF snake in handling boundary concavities, wide capture range, and its insensitivity to initialization. Using GVF snake as the main segmentation method and assist with user interface, we can properly segment the confocal microscopic insect brain image for most of the cases. For those images that are ill segmented, we need better preprocessing techniques for more satisfactory result. For the others, GVF snake is an acceptable method for image segmentation.

REFERENCES

1. M. Kass, A. Witkin, and D.Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, pp.321-331, 1987.
2. D. Terzopoulos and K. Fleischer, "Deformable models," *Vis. Comput*., vol. 4, pp. 306-331, 1988.
3. F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Trans. Pattern Anal. Machine Intell*., vol. 15, pp. 617-634, 1993.
4. D. Terzopoulos and R. Szeliski, "Tracking with Kalman snakes," in *Active Vision*, A. Blake and A. Yuille, Eds. Cambrige, MA: MIT Press, 1992, pp. 3-20.
5. Chenyang Xu and Jerry L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Transactions on Image Process*, vol. 7, no. 3, Mar. 1998.
6. S. N. Gupta and J. L. Prince, "Stochastic models for DIV-CURL optical flow methods," *IEEE Signal Processing Lett.*, vol. 3, pp. 32-35, 1996.
7. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Interscience, 1953.
8. S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," *IEEE Trans, Med. Imag*., vol. 14, pp. 12-24, Mar. 1995.
9. H. Freeman, "On the encoding of arbitrary geometric configuration," *IRE Transactions on Electronic computers*, 1961.
10. T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*. MIT Press, 1989.
11. A. H. Charles and T. A. Porsching, *Numerical Analysis of Partial Differential Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1990.
12. W. F. Ames, Numerical Methods for Partial Differential Equations, 3rd ed. New York: Academic, 1992.