Fluid Vector Flow and Applications in Brain Tumor Segmentation

Tao Wang, Irene Cheng and Anup Basu *

Abstract—**We propose a new approach, that we call "Fluid Vector Flow" (FVF) active contour model to address problems of insufficient capture range and poor convergence for concavities. With the ability to capture a large range and extract concave shapes, FVF demonstrates improvements over techniques like GVF, BVF and MAC on three sets of experiment - synthetic images, pediatric head MRI images, and brain tumor MRI images from the Internet Brain Segmentation Repository.**

Index Terms—**Active contour models, vector flow, snakes, brain tumor, segmentation**

I. Introduction

Active contour models or snakes [1-5] have been adopted as effective tools for segmentation [6-7] and object tracking [8-9]. Active contour models discussed in the literature can be classified into two categories: parametric [1-3] and level set [4-5].

Parametric active contour models are represented explicitly as polynomials or splines. Given an initial contour, the evolution of a parametric active contour model is driven by external forces while the shape of the contour is maintained by the internal forces [1]. Due to the availability of efficient numeric methods [1-2], parametric active contour models are often faster than level set ones [4-5]. Given a single initial contour as the input, parametric active contour models are able to extract a single object. Despite the above strength, parametric snakes have two weaknesses. First, the capture range is limited. Capture range is the region that the external forces are strong enough to drive contour evolution. The external forces of the traditional [1] and Gradient Vector Flow (GVF) [2] parametric snakes are represented as small arrows in Fig. 1 (a) and (b). The length of an arrow represents the magnitude of an external force at that location. In Fig. 1, the capture range is the region with dense arrows (external forces) that are strong enough to drive the contour evolution. We can see that the capture range of the traditional snake is a very limited region around the object boundary. GVF diffuses the external forces from the object boundary to its surroundings to obtain a larger capture range. However, the capture range of GVF is still not the entire image. If the initialization is out of the capture range, the active contour will not evolve (Fig. 1 (c)). Second, parametric snakes, e.g. GVF and Boundary Vector Flow (BVF) [3], are unable to extract acute concave shapes (Fig. 2 (b) and (c)). We observe that other parametric active contour models (traditional, GVF and BVF) are unable to extract acute concavities because their external force fields are static. There could be saddle points or stationary points [4] where the composition of external forces is zero (Fig. 2 (d) and (e)) in static force fields. Therefore, the contours will get stuck at those locations and equilibrium will be achieved too early [4].

Level set active contour models [4-5] are implicitly represented in the zero level set. The evolution of a level set active contour model is achieved by deforming the level set function. The advantages of level set include the abilities to capture multiple objects and complex geometries [4]. However, they are usually slower than parametric methods because the deformation of a higher dimensional function is required [4]. Morse et al. [12] proposed to implicitly represent snakes using radial basis functions by placing them at some landmarks. This can avoid manipulating a higher dimensional function but it requires insertion and deletion of landmarks dynamically. Moreover, "false" objects can be extracted in the presence of noise which may cause multiple zero level sets (Fig. 3).

For many applications [10], it is necessary to extract a single object. Since the application in this paper is to extract single objects, i.e., the brain tumor, in brain MRI images, we focus on parametric snakes taking advantage of its uniqueness of contour. To address the issues of capture range and acute concave shape, we propose a parametric active contour model using a new concept, that we call "Fluid Vector Flow" (FVF), which simulates fluid flowing along object boundary and generates external force fields dynamically to drive the contour evolution. FVF has the largest capture range, i.e., the entire image. FVF is also able to extract acute concave shapes due to its non-static external force fields. In this model, the external force field changes dynamically with the contour evolution. Thus, the FVF contour does not get stuck and acute concavities can be extracted.

The limitations with previous models that FVF will overcome are as follows:

Manuscript received September 25, 2008; revised December 22, 2008. This work was supported in part by AHFMR, iCORE and NSERC.

Tao Wang is with the Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2H1 (e-mail: taowang@cs.ualberta.ca).

Irene Cheng is with the Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2H1 (e-mail: lin@cs.ualberta.ca).

* Anup Basu is with the Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2H1 (corresponding author, phone: 780-492-3330; fax: 780-492-1071; e-mail: anup@cs.ualberta.ca).

- Limited capture range
- Inability to handle acute concave shapes

Extraction of brain tumor in brain MRI images becomes much easier by taking advantage of the proposed FVF method.

The rest of this paper is organized as follows. In Section II, the background of parametric and level set snakes is briefly described. Section III introduces the Fluid Vector Flow method. Some experimental results and comparisons are presented in Section IV, before the work is concluded in Section V.

II. Background of Parametric and Level Set Active Contour Models

A. *Traditional snake*

A traditional snake [1] is a parametric active contour:

$$c(s) = (x(s), y(s)), s \in [0,1] \tag{1}$$

Given an initial contour, it evolves within an image $I(x, y)$ to minimize the energy function:

$$E_{snake} = \int_0^1 [E_i(c(s)) + E_e(c(s))]ds \tag{2}$$

where E_i is the internal (spline) energy and E_e is the external energy.

The internal energy is given by:

$$E_i = \frac{\alpha(s)\,|\,c'(s)\,|^2 + \beta(s)\,|\,c''(s)\,|^2}{2} \tag{3}$$

In many implementations, the coefficient of the first-order term in (3) is a constant, *α(s) = α*; and *β(s)* is set to zero to allow the snake to be second-order discontinuous and contain corners. Many parametric snakes share the same internal energy. They differ mostly in the external energy. A snake should evolve to minimize the energy functional E_{snake}. This problem can be formulated with the Euler-Lagrange equation. In calculus of variations, the Euler-Lagrange equation of:

$$J[c(s)] = \int_{s_0}^{s_1} F(s, c(s), c'(s), c''(s))ds \tag{4}$$

is represented by:

$$F_c - \frac{d}{ds}F_{c'} + \frac{d^2}{ds^2}F_{c''} = 0 \tag{5}$$

Therefore, the Euler-Lagrange equation of (2) is represented by:

$$\alpha c''(s) - \beta c''''(s) + \nabla E_e = 0 \tag{6}$$

To find a numeric solution of (6), the snake is treated as a function of time *t* as well as *s*:

$$\alpha c''(s,t) - \beta c''''(s,t) + \nabla E_e = 0 \tag{7}$$

When the contour stabilizes, the time term vanishes and a solution is obtained.

B. *GVF snake*

GVF snake [2] has a larger capture range than the traditional snake. It diffuses the edge information from the object contour to its neighborhood. The external force of GVF snake differs from the traditional snake in that it cannot be written as the negative gradient of a potential function. In addition to this, the GVF snake is formulated directly from a force balance condition rather than a variational formulation. The gradient vector flow is defined as the vector field:

$$G_{gvf}(x, y) = (u(x, y), v(x, y)), x \in [x_0, x_1]\, and\; y \in [y_0, y_1] \tag{8}$$

that minimizes the energy function:

$$E_{gvf} = \int_{y_0}^{y_1}\int_{x_0}^{x_1} (k(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \left|\nabla f\right|^2 \left|G_{gvf} - \nabla f\right|^2)dxdy \tag{9}$$

where *k* is a blending parameter, u_x, u_y, v_x, *and* v_y are the derivatives of the vector field, and ∇f is the gradient of the edge map. The GVF snake is computed by solving the following Euler-Lagrange equations:

$$k\nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \tag{10}$$

$$k\nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \tag{11}$$

C. *BVF snake*

BVF [3] extends the capture range further to the entire image based on interpolation. It applies a threshold to generate a binary boundary map of the input image. Then, four potential functions Ψ_x , Ψ_y , Ψ_{xy} and Ψ_{yx} are computed using line-by-line interpolations in the horizontal, vertical and two diagonal directions. The boundary vector flows are defined based on the gradients of the following potential functions:

$$\Phi_1 = (\nabla\Psi_x, \nabla\Psi_y) \tag{12}$$

$$\Phi_2 = (\frac{\sqrt{2}}{2}(\nabla\Psi_{xy} + \nabla\Psi_{yx}), \frac{\sqrt{2}}{2}(\nabla\Psi_{xy} - \nabla\Psi_{yx})) \tag{13}$$

The external force is defined as:

$$E_e(x, y) = \Phi(x, y) \tag{14}$$

Similar to GVF, BVF is unable to extract acute concavities.

D. Magnetostatic Active Contour (MAC) Model

MAC snake [4] is a level set active contour model. The external force of MAC is based on magnetostatics and hypothesized magnetic interactions between the active contours and object boundaries. It is able to capture complex geometries and multiple objects with a single initial contour. However, as stated in the introduction, it is slower than parametric methods and may detect multiple false objects in the presence of noise, which may cause multiple zero level sets to arise. MAC snake represents the active contour with an implicit model in which the contour consists of all points in:

$$c=\{\bar{x}\mid\phi(\bar{x})=0\},\ \text{where}\ \phi:R^2\rightarrow R \tag{15}$$

MAC relates the motion of that contour to a PDE (Partial Differential Equation) on the contour:

$$\frac{\partial\phi}{\partial t}=-\nabla\phi\cdot\bar{v}(t) \tag{16}$$

where $\bar{v}(t)$ describes the velocity of the contour movement. For image segmentation:

$$\frac{\partial\phi}{\partial t}=\alpha s(\bar{x})\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)|\nabla\phi|-(1-\alpha)F(\bar{x})\cdot\nabla\phi \tag{17}$$

where α is a real constant, $s(\bar{x})$ is the stopping function, and $F(\bar{x})$ is the magnetostatic force.

III. Proposed Fluid Vector Flow (FVF) Method

Given an input image *I(x, y)* and a closed parametric contour *c(s)* given in (1), the objective is to evolve the contour to extract a target object *O(x, y)*, i.e. the brain tumor. This method is executed in three stages: binary boundary map generation, vector flow initialization and fluid vector flow computation. Fig. 4 shows the flowchart of this method, which starts with initializing the contour, then the binary boundary map is generated and vector flow is initiated, and fluid vector flow is computed and dynamically updated until the object contour is extracted.

In the first stage, we apply a Gaussian smoothing filter to the input image and apply a gradient operator to find edges in the image. A threshold (free parameter) $T\in[0, 1]$ is then used to generate the binary boundary map. At the second stage, the contour can be initialized to initialize the external force field. The initial contour can be inside, outside, or overlapping the target object. The program automatically detects the initialization and generates the external force field accordingly. The computation of the internal energy follows (3). The initial forces will push the active contour to the neighborhood of the target object. At the last stage, a control point is automatically selected from the object boundary and generates new external force field to evolve the active contour. This point can flow freely along the object boundary like a drop of fluid, dynamically update the external force field to avoid the problem of saddle points and stationary points [4], and therefore further evolve the active contour until convergence is achieved.

A. Binary Boundary Map Generation

The boundary map is defined as:

$$M_B(x,y)=|\nabla(-G_\sigma(x,y)*I(x,y))| \tag{18}$$

where $G_\sigma(x, y)$ is a Gaussian smoothing filter with standard deviation σ, $*$ is the convolution operator and ∇ is the gradient operator. We compute the normalized boundary map:

$$M_{NB}(x,y)=\frac{M_B(x,y)-\min(M_B(x,y))}{\max(M_B(x,y))-\min(M_B(x,y))} \tag{19}$$

Similar to BVF [3], we apply a threshold $T\in[0, 1]$ to generate the binary boundary map:

$$M_{BB}(x,y)=\begin{cases}1, & \mathit{if}\ M_{NB}(x,y)>T\\ 0, & \mathit{otherwise}\end{cases} \tag{20}$$

The choice of a suitable threshold value varies depending on the intensity distribution and contrast associated with the set of images being analyzed. For the brain MRI images tested in our implementation, a default value of 0.1 works well. Observe that the blurred contour of the brain ventricle (low intensity region) is extracted successfully in the boundary map using $T = 0.1$ (Fig. 5). We tested with a higher T value and then decreased T progressively but object boundary continuity was not obtained until 0.1 was reached. The extracted boundary provides an envelope to ensure that the final convergence is not out of bound. The threshold of 0.1 is consistent with the threshold of 0.13 commonly used by other snake models as suggested by Sum and Cheung in [3].

B. Vector Flow Initialization

At this stage, the contour should be initialized to initialize the external force field. The initial parametric contour *c(s)* can be initialized either inside, outside, or overlapping (Fig. 6 (a)-(c)) the target object *O(x, y).* The FVF method is insensitive to the initialization by taking advantage of the binary boundary map generated at the previous stage. Suppose C is the initial contour, R_c is the region enclosed by contour C, and R_b is the region enclosed by the binary boundary map (Fig. 7), we define $R_{bc}:=R_b\cap R_c$. The following criteria determine the spatial relationships between the initial contour and the binary boundary map:

(a) C is inside the binary boundary map when $R_{bc}=R_c$;

(b) C is outside the binary boundary map when $C\cap R_{bc}=\phi$;

(c) Otherwise, C is overlapping the binary boundary map.

When overlap is detected, contour C will be automatically enlarged guided by the boundary map to enclose R_b. Therefore, “overlapping” is eventually handled as “outside” after enlargement (Fig. 6 (d)). During enlargement, neighboring objects in the binary boundary map that do not contain R_{bc} remain outside the contour and only the one connected component (the target) in the binary boundary map that contains R_{bc} is included in R_c. A connected component is a region of 8-connected object pixels (ones) in the binary boundary map.

We use the discrete form of (1) to represent the contour C:

$$c(i) = \{(x_i, y_i)\}, i \in [0,1,\ldots,P-1] \qquad (21)$$

where P is the number of points on the contour.

The center point of the bounded region is located at:

$$(x_c, y_c) = (\sum_{i=0}^{P-1} x_i / P, \sum_{i=0}^{P-1} y_i / P) \qquad (22)$$

An external energy function is defined as:

$$E_e(x,y) = \begin{cases} \chi(f_x + \delta\cos\phi, f_y + \delta\sin\phi) \ when \ M_{BB}(x,y) = 0 \\ 0 \qquad otherwise \end{cases} \qquad (23)$$

where χ is a normalization operator, $\delta = \pm 1$ (controls the inward or outward direction, when the contour is “outside” or “inside”), $(f_x, f_y) = \chi(\nabla I(x,y))$, and:

$$\phi = \begin{cases} \arctan(\frac{y-y_c}{x-x_c}) & when \quad x \neq x_c \\ \frac{\pi}{2} & when \quad x = x_c \ and \ y > y_c \\ \frac{3\pi}{2} & when \quad x = x_c \ and \ y < y_c \end{cases} \qquad (24)$$

where $\phi \in [0, 2\pi]$.

The external energy E_e has a gradient component and a directional component. The gradient force is computed in a manner similar to the traditional snake [1] and GVF snake [2]. The characteristic of FVF lies in the computation of the directional force, which is based on a polar transformation. When the contour is far away from the object, the directional force dominates and attracts the contour towards the object. When the contour is close to the object, the gradient force fits the contour to the object.

The capture range of FVF extends to the whole image because the vector flow energy defined in (23) spreads around the entire image *I(x, y)*. Even if the initial contour is far from the object, the snake can still evolve towards the object. In other words, the border of an image can be used as the initial contour when the initial contour is not given. This feature makes FVF more effective than either the traditional snake [1] or the GVF snake [2]. Although the capture range of BVF [3] can also extend to the entire image, the performance of FVF is more efficient because interpolation is avoided. Furthermore, the BVF interpolation is executed in only four directions, whereas FVF is direction invariant ($\phi \in [0, 2\pi]$). Note that the capture range of MAC is also the entire image. Fig 10 (p) – (t) illustrates the capture ranges of GVF, BVF, MAC and FVF. Fig 10 (p) shows an object without a given initial contour. The image border is then used as the initial contour. GVF failed to extract the object since the initial contour (image border) is out of its capture range. BVF, MAC and FVF can extract the object since their capture ranges cover the entire image domain.

Note that if concavities exist, convergence will not be achieved at the vector flow initialization step. Fig. 8 (b) shows the vector flow initialization. The circle is the initial contour and the blue (dark gray in B&W print) small arrows represent vector flows. Fig. 8 (c) shows that the evolution stops at the center of the image where the composition of external forces is zero. The contour will evolve to the red (middle grey in B&W print) line and is not able to extract the concave region. To extract the complete contour, the fluid vector flow computation step is performed.

C. Fluid Vector Flow Computation

In this step, a trace method is applied to the binary boundary to get a list of control points:

$$B(x_q, y_q) = \xi(M_{BB}(x,y)), q \in [0,1,\ldots,Q-1] \qquad (25)$$

where ξ is a boundary trace operator and Q is the number of the control points.

The Fluid Vector Flow energy function is defined by:

$$E_{fvf}(x,y) = \begin{cases} \chi(\chi(\nabla \mathrm{I(x,y)}) + \delta\chi(x - x_q, y - y_q)), \\ \quad if \ M_{BB}(x,y) = 0 \ \ \& \ \ (x_q, y_q) \in B \\ 0 \quad otherwise \end{cases} \qquad (26)$$

The pseudo code of Fluid Vector Flow computation and active contour evolution is as follows:

```
0  Get a list of control points B with (25). B contains Q points.
1  ind=0;
2  While convergence is not achieved
3      q=ind+δ;
4      if q>=Q
5          break;
6      x=B[q].x; y=B[q].y;
7      Generate new force field at (x, y) with (26);
8      Evolve the active contour in the new force field;
9  End while
10     Output the result of contour evolution.
```

In the pseudo code, *ind* (line 1) is the index of control point, *q* (line 3) is the new index of the control point, and (*x*, *y*) (line 6) is the new location of the control point.

Our intention is to use the control points to generate the external force fields. First, a list of control points *B* is computed with (25). Then, in each iteration of the above while loop, one control point is sequentially selected in the list *B*. Imagine that the control point is a drop of fluid, this process looks like the fluid drop moving along the object boundary and generating vector flow (external force field) dynamically. This is the reason why we name the method Fluid Vector Flow.

Using all the control points to generate force fields can be time-consuming. In addition to that, adjacent control points may generate external force fields with little differences. Therefore, a parameter δ is used to manage the selection of the control point. δ can be imagined as the velocity of the movement of the fluid drop. The method selects 1 out of δ control points to achieve better time efficiency. For instance, in Fig. 8, we assign δ =20 to select one out of 20 control points. When δ=1, all points extracted by the trace operator are used one by one.

Once the control point moves to its new location (i.e., a new control point is selected), it generates new external force field to avoid the problem incurred by saddle points and stationary points, and therefore is able to further evolve the contour until convergence is achieved.

The energy E_{fvf} has a gradient term and a directional term. The directional force attracts the evolving contour towards the control points even for control points in a concave region. When the contour is close to the object, the gradient force fits the contour onto the object. Convergence is achieved when the contour stops evolving.

In Fig. 8, the target object is given in (a), and (b) shows the initial contour and vector flow initialization. From (c) to (k), the fluid vector flows are shown as blue (dark grey in B&W print) arrows, and the evolving contour is illustrated in red (middle gray in B&W print). The control point involved in each convergence step is marked as a green (light gray in B&W print) dot. Note that the control points selected by the boundary trace operator are located along the binary boundary map.

In Fig. 9, we show the results of applying our technique on a head MRI image to extract the low intensity brain ventricle region.

IV. Experimental results

We tested and compared FVF against GVF, BVF and MAC for three data sets: synthetic images, head MRI images, and IBSR brain tumor MRI images [13]. The head MRI images were provided by the Department of Pediatrics at the University of British Columbia. The IBSR brain tumor MRI images were provided by the Center for Morphometric Analysis at Massachusetts General Hospital, and are available at http://www.cma.mgh.harvard.edu/ibsr/. These T1-weighted images contain multiple scans of a patient with a tumor taken at roughly 6 month intervals over three and a half years. The dimension of the images is 256x256. The pixel resolutions on these images are 0.9375x0.9375 mm in-plane by 3.1 mm slice thickness. Segmentation ground-truth files are also available for quantitative analysis.

We also set up parameters for those snake models to compare them as fairly as possible. For GVF, we keep the default settings unchanged. For BVF, we test the input images with 9 different values (0.1, 0.2, …, 0.9) of the threshold *T* and report the best result. For MAC snakes, in addition to these 9 threshold values, we also tested it with 2 much smaller values (0.01 and 0.05) according to the suggestion of the authors [4]. Moreover, since dual snake contours (Contour 0 and Contour 1) are implemented in MAC, we report the contour that has better result. For FVF snakes, we use the threshold value determined by BVF.

For each test image, the initial contour was placed inside, outside and overlapping the target object to test the robustness and sensitivity of the methods to initializations. Since the 3rd test set (IBSR brain tumor images) comes with noise, we did not artificially add noise to the data.

A. Synthetic Images

We first tested and compared FVF with GVF, BVF and MAC snakes for a set of synthetic images. Some results are shown in Fig. 10.

The 1st row (Fig. 10 (a)-(e)) shows an acute concave object with an initial contour at the outside, and the result of GVF, BVF, MAC (Contour 1) and FVF. We can see that both MAC and FVF can extract the boundary of the object. However, GVF and BVF failed to do so. This is because both GVF and BVF are incapable of extracting acute concave shapes.

The 2nd row (Fig. 10. (f)-(j)) shows an object with a small initial contour inside, and the results of GVF, BVF, MAC (Contour 0) and FVF. We can see that the GVF snake does not evolve at all because of the static equilibrium force field. BVF, MAC, and FVF can extract the boundary of this object.

The 3rd row (Fig. 10. (k)-(o)) shows an object with an overlapping initial contour, and the result of GVF, BVF, MAC (Contour 0 and Contour 1) and FVF. We can see that the GVF and BVF snakes did not extract the boundary of the object but instead evolved to a point. Each one of the dual contours of MAC snake superimposes on each other and outlines both the internal and external boundaries of the object. The FVF snake extracts the boundary of the object.

The 4th row (Fig. 10. (p)-(t)) shows an object without a given initial contour (the image border is then used as the initial contour), and the results of GVF, BVF, MAC (Contour 1) and FVF. GVF failed to extract the object since the initial contour (image border) is out of its capture range. BVF, MAC and FVF can extract the object since their capture ranges cover the entire image domain.

B. Head MRI Images

We then tested and compared FVF with GVF, BVF and MAC snakes for a set of head MRI images. Some results are shown in Fig. 11.

The first row (Fig. 11 (a)-(e)) shows an image with an initial contour outside the high intensity region (intra-ventricular hemorrhage), and the result of GVF, BVF, MAC (Contour 1) and FVF. We can see that only BVF and FVF can extract the boundary of the object.

The second row (Fig. 11. (f)-(j)) shows an image with a small initial contour inside the brain ventricle, and the result of GVF, BVF, MAC (Contour 1) and FVF. We can see that the GVF and BVF snakes evolved to lines on the right side of the brain ventricle but failed to extract the boundary of the ventricle. MAC and FVF can both extract the boundary of the brain ventricle. However, the result of FVF is smoother.

The third row (Fig. 11. (k)-(o)) shows an image with an initial contour overlapping the eye, and the result of GVF, BVF, MAC (Contour 1) and FVF. We can see that the GVF and BVF snakes degenerated to points on the top right and top left of the eye. MAC and FVF can both extract the eye and the results are similar.

Two examples of visual evaluation comparing the manually defined contour and the FVF generated contour are shown in Fig. 12. Quantitative evaluations of GVF, BVF, MAC and FVF are reported in the next section.

C. IBSR Brain Tumor MRI Images and Quantitative Analysis

Brain tumor images (in which brain tumors are visible) from IBSR [13] are tested in our experiment. The Tanimoto Metric [11] is used for quantitative analysis.

Tanimoto Metric is defined as:

$TM = \frac{\|R_X \cap R_G\|}{\|R_X \cup R_G\|}, 0 \le TM \le 1$, where R_X is the region enclosed by the contour generated by the test method, R_G is the region enclosed by the ground-truth contour, and $\|\cdot\|$ is set cardinality (number of elements). $TM = 0$ would indicate two completely distinct contours; while $TM = 1$ would indicate completely identical contours. Table I shows the test results.

Table I has 5 columns. The 1st column is the image ID (1 to 10). The 2nd column lists the name of the 4 methods. The 3rd to 5th columns are the Tanimoto Metric when the initial contour is inside, outside and overlapping the target object. The best method for each image is bold. FVF outperforms GVF, BVF and MAC in general.

It is important to note that the four methods apply different computational models to generate the force fields which are governed by the underlying image properties. Force fields are unevenly distributed in an image. In other words, a method may perform well in one region but may not do well in another region of the same image. An example is illustrated in Fig. 13, which shows the results of GVF, BVF, MAC and FVF on test image #4 when the initial contour (not shown) is inside (2nd and 3rd rows) and outside (4th and 5th rows) the brain tumor respectively. Observe that when the initial contour (not shown) is inside the tumor, the Tanimoto Metric (TM) value of MAC is the best among the four methods (0.876, see Table 1). However, when the initial contour (not shown) is outside the tumor, the TM value of MAC is the worst (0.080, see Table 1). Also note that a perfect TM value of 1.0 is difficult to achieve especially when the target object is small. For example, the TM value is only 0.876 even though the method generated contour and the ground-truth are similar (see (b) and (i) in Fig. 13). This is because the brain tumors are small, composed of only 50 to 200 pixels in the MRI images; a few pixels of deviation from the ground-truth can result in a less than perfect TM value.

Table II shows the mean, median and standard deviation of the TM of GVF, BVF, MAC, and FVF. FVF has the largest mean and median with smallest standard deviation. Mixed effects model [14] with random data and test effects is used to statistically compare the differences between FVF and other 3 methods. Random data effects are used because of the induced correlation due to the tests on the same data set, and random test effects because of the repeated tests (i.e., inside, outside, and overlap) for each method on the same data sets. Table III shows the comparison between FVF and GVF, BVF, and MAC using mixed effects model. Since all the P-values are smaller than 0.01, FVF is statistically better than the other models in this experiment.

D. Implementation

The MATLAB source codes of GVF and BVF are obtained from the authors of [2] and [3]. The executable java code of MAC is provided by the authors of [4]. FVF is implemented in MATLAB. In our implementation, the default value of T is 0.1. The program also provides the binary boundary maps generated by 8 other values (0.2, 0.3, …, 0.9) of T so that the software user can choose the best value to define the edges of the target object, if the default value does not work. FVF is implemented in MATLAB and not optimized for speed. In general, it takes FVF about 1 to 5 seconds to process a 256x256 image on a Pentium 4 (3GHz CPU, 2GB RAM) desktop computer.

V. Conclusion and future work

We proposed a new parametric Fluid Vector Flow (FVF) active contour model to address the issues of limited capture range and the inability to extract complex contours with acute concavities. Experiments on synthetic images and head MRI images show that FVF produces better results compared to GVF, BVF and MAC. Quantitative experiments on brain tumor images show that FVF has the largest mean (0.61) and median (0.60) with smallest standard deviation (0.05) using Tanimoto Metric. Mixed effects model with random data and test effects is used to statistically compare the differences between FVF and other three methods. Since all the P-values are smaller than 0.01, FVF is statistically better than the other models in this experiment. In future work, we plan to extend FVF to analyze 3D medical data.

Acknowledgment

The authors thank Dr. Steven Miller, University of British Columbia, for providing the infant brain MRI data used in this work. The authors thank Dr. Shoo Lee and Mr. George Qiaohao Zhu, University of Alberta, for discussion on the infant brain MRI processing and statistical analysis methods. We also want to thank the Center for Morphometric Analysis at Massachusetts General Hospital for providing the brain tumor MRI images.

References

[1] M. Kass, A. Witkin, and D. Terzopoulus, “Snakes: Active contour model”, *Intl. J. of Computer Vision*, vol. 1(4), pp. 321-331, 1988.

[2] C. Xu and J. L. Prince, “Snakes, shapes, and gradient vector flow”, *IEEE Trans. on Image Processing*, pp. 359-369, 1998.

[3] K.W. Sum and P. Y. S. Cheung, “Boundary vector field for parametric active contours”, *Pattern Recognition*, pp. 1635-1645, 2007.

[4] X. Xie and M. Mirmehdi, “MAC: Magnetostatic Active Contour Model”, *IEEE Trans. PAMI*, vol. 30(4), pp. 632-645, 2008.

[5] O. Juan, R. Keriven, and G. Postelnicu, “Stochastic motion and the level set method in computer vision: Stochastic active contours”, *Intl. J. of Computer Vision*, vol. 69(1), pp. 7–25, 2006.

[6] I. Dagher and K. E. Tom, Water, “Balloons: A hybrid watershed Balloon Snake segmentation”, *Image and Vision Computing*, vol. 26 (7), pp. 905-912, 2008.

[7] S. W. Yoon, C. Lee, J. K. Kim and M. Lee, “Wavelet-based multi-resolution deformation for medical endoscopic image segmentation”, *J. of Medical Systems*, vol. 32 (3), pp. 207-214, 2008.

[8] B. H. Lee, I. Choi and G. J. Jeon, “Motion-based boundary tracking of moving object using parametric active contour model”, *IEICE Trans. on Info. and Sys.*, E90-D (1), pp. 355-363, 2007.

[9] M. Li and C. Kambhamettu, “Automatic Contour Tracking in Ultrasound Images”, *Clinical Ling. and Phon.*, vol. 19 (6-7), pp. 545-554, 2005.

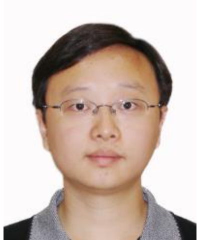
[10] L. He, Z. Peng, B. Everding, X. Wang, C. Y. Han, K. L. Weiss, and W. G. Wee, “A comparative study of deformable contour methods on medical image segmentation”, *Image and Vision Computing*, vol. 26, pp.141-163, 2008.

[11] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, USA: Academic Press, 1999, p. 366.

[12] B. Morse, W. Liu, T. Yoo, and K. Subramanian, “Active contours using a constraint-based implicit representation”, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005, pp. 285–292.

[13] http://www.cma.mgh.harvard.edu/ibsr/

[14] J. D. Singer, “Using SAS PROC MIXED to Fit Multilevel Models, Hierarchical Models, and Individual Growth Models”, *J. of Educational and Behavioral Stat.*, vol. 24, pp. 323-355, 1998.



Tao Wang received B.Eng. and M.Eng. from the Department of Computer Science and Engineering, Zhejiang University, Hangzhou, Zhejiang Province, China. He is currently a Ph.D. candidate in the Department of Computing Science at the University of Alberta, Edmonton, Alberta, Canada.

He worked as a software engineer in the broadband department at UTStarcom, Hangzhou, Zhejiang Province, China, from 2003 to 2004. His current research interests include active contour techniques and 3D skeletonization. He has 7 publications in international conferences and journals.

Mr. Wang has won a number of international, national, provincial, institutional, and departmental awards for his academic achievement.

Irene Cheng is the Chief Scientist of the Online Multimediaucation Project sponsored by iCORE and Castle Rock Research and the 4D Medical Imaging project sponsored by Alberta Science and Research in the Department of Computing Science at the University of Alberta, Edmonton, AB, Canada. She has over ten years of R&D experience in industry, including Lloyds Bank and Zoomage, working on various information technology and 3-D vision systems. One of her unique contributions is to incorporate human perception Just-Noticeable-Difference (JND)—with scale-space analysis, systematically based on psychophysical methodology, to improve 3-D simplification and transmission techniques. By applying the JND approach, automatic selection of scale based on a given object dimension is possible. Her research includes multimedia modeling, visualization and transmission, perceptual issues, and multimedia database structure. Her current research also covers optimizing transmission over unreliable networks taking packet loss into consideration. She has over 40 publications in international conferences and journals.

Dr. Cheng has received numerous Scholarships and Fellowship from NSERC and iCORE among others.

Anup Basu received the B.S. degree in mathematics/statistics and the M.S. degree in computer science from the Indian Statistical Institute. He received the Ph.D. degree in computer science from the University of Maryland, College Park.

He was with Tata Consultancy Services, New Delhi, India, and Biostatistics Division, Strong Memorial Hospital, Rochester, NY. He is currently a Professor at the University of Alberta, Edmonton, AB, Canada. He is also an iCORE—Castle Rock Research—SUN Industrial Research Chair in Multimedia. He has published over 130 papers, patents, books and book chapters, in leading conferences and journals. He has also helped startup several technology companies. His current research interest includes Computer Vision, Graphics, 3-D Multimedia and Multimedia Communications over Heterogeneous Networks. He pioneered the use of foveation in image and video compression and stereo visualization in 1992–1993. He also introduced several new panoramic image sensors in the mid-1990s. These research directions have been subsequently pursued by many researchers in leading institutes around the world.

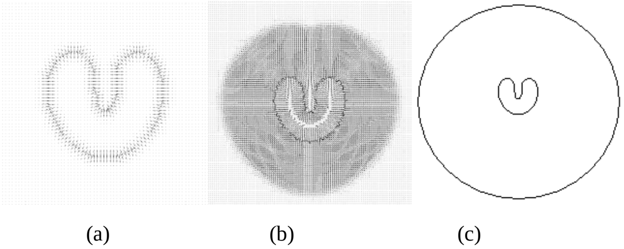


Fig. 1. The limited capture range of (a) a traditional parametric snake and (b) a GVF parametric snake. If the initialization (outer circle in (c)) is outside the capture range, convergence does not occur.

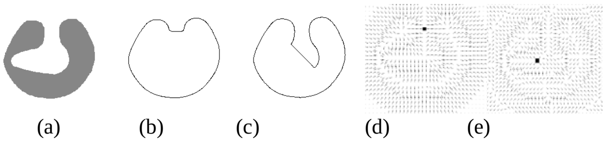


Fig. 2. (a) An acute concave shape. (b) GVF and (c) BVF are not able to capture the acute concave shape. A saddle point in GVF is shown in (d) and a stationary point in BVF is shown in (e).

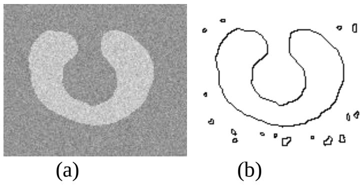


Fig. 3. (a) A “U-shape” object in noisy environment (b) false objects (i.e., small enclosed contours) can be extracted by a level set snake.

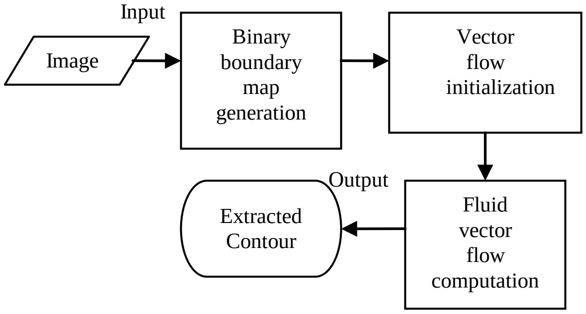

Input
Image
Binary boundary map generation
Vector flow initialization
Output
Extracted Contour
Fluid vector flow computation

Fig. 4. The process of FVF.

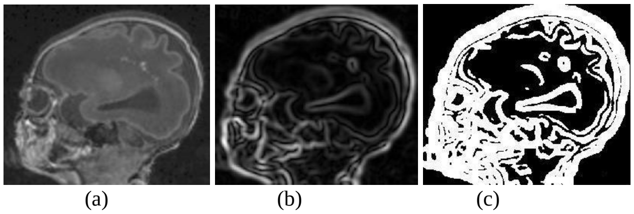


Fig. 5. (a) A head MRI image, (b) its gradient map and (c) its extracted boundary map using a default threshold of 0.1.

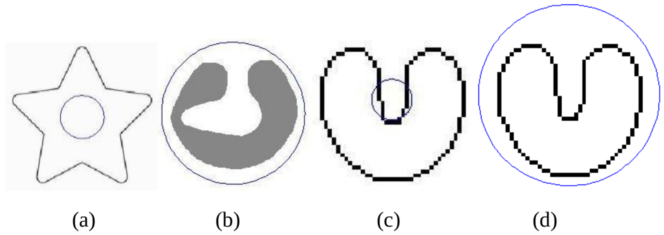


Fig. 6. The initial contour (circle) is (a) inside (b) outside and (c) overlapping the target object.(d) the initial contour is automatically enlarged to enclose the object so that "overlapping" can be handled as "outside."

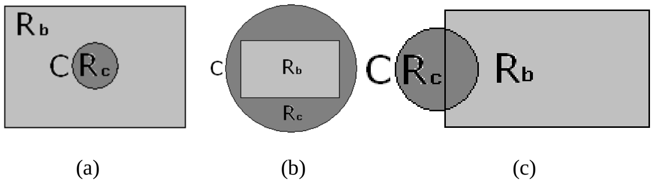


Fig. 7. (a) Initial contour C is inside R_b , (b) contour C is outside R_b , and (c) contour C overlaps R_b . FVF is able to evolve in each of these initialization cases.

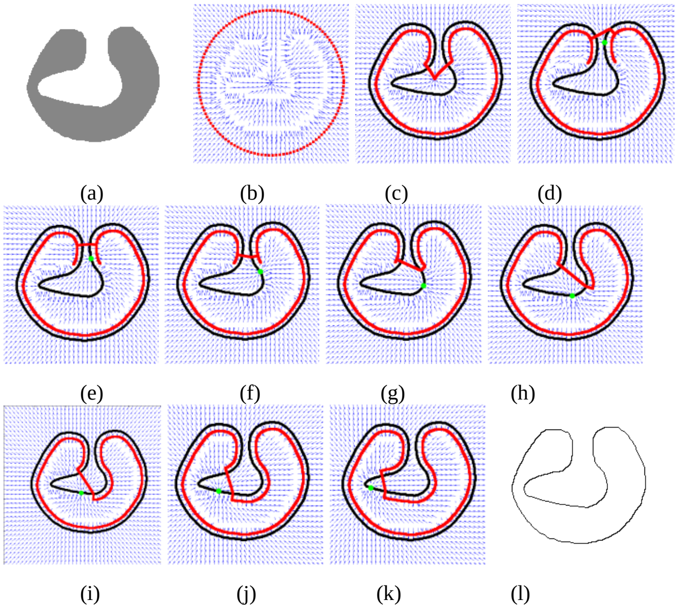


Fig. 8. An example of FVF contour evolution: (a) The target object and (b) the initial contour and vector flow initialization, (c)-(k) a sequence of fluid vector flow processes and (l) the convergence result.

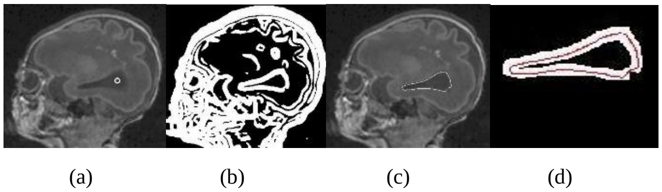


Fig. 9. Illustration of FVF process: (a) the target object (brain ventricle) with initial contour (small circle in the ventricle) added, (b) the binary boundary map, (c) the final contour of FVF in the image, and (d) a zoomed-in view of the binary boundary map which restricts the final contour inside an envelop.

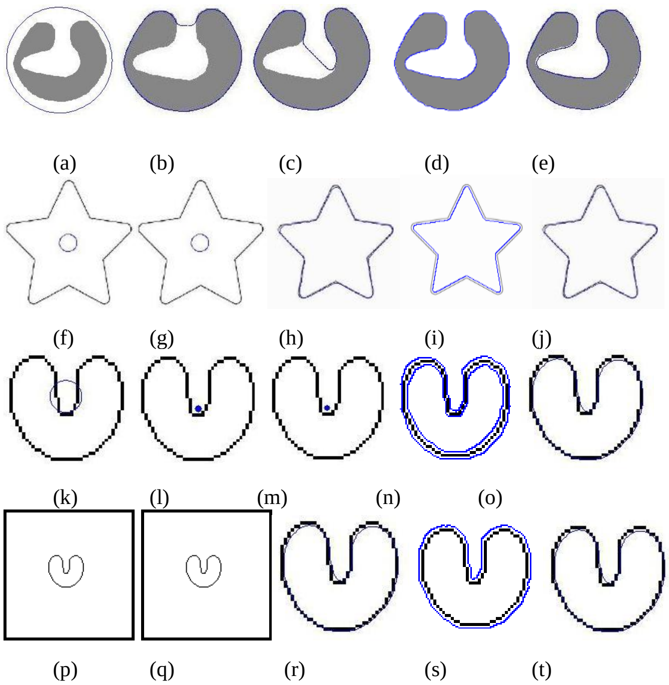


Fig. 10. (a) An acute concave object with an initial contour at the outside, and the results of: (b) GVF, (c) BVF, (d) MAC (e) FVF; (f) an object with a small initial contour at the inside, and the results of: (g) GVF, (h) BVF, (i) MAC (j) FVF; (k) an object with an overlapping initial contour, and the results of (l) GVF, (m) BVF, (n) MAC (o) FVF; (p) an object with the image border as the initial contour, and the results of (q) GVF, (r) BVF, (s) MAC (t) FVF.

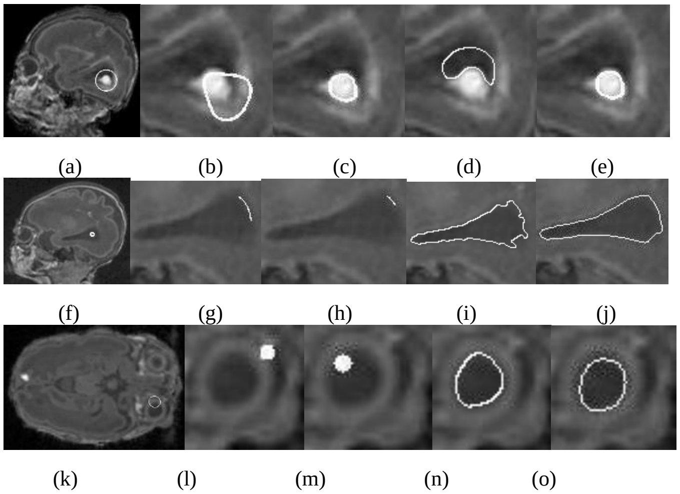
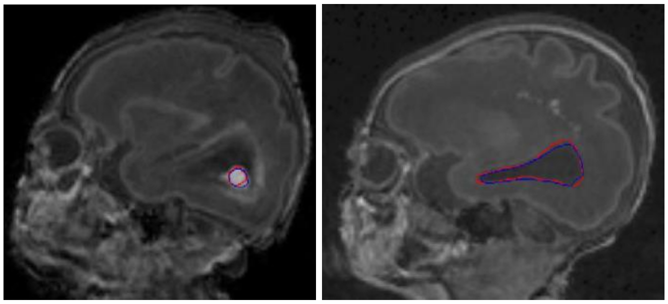


Fig. 11. (a) An image with an initial contour on the outside of the high intensity region (intra-ventricular hemorrhage), and the results (zoomed-in) of: (b) GVF, (c) BVF, (d) MAC (e) FVF; (f) an image with a small initial contour at the inside of the brain ventricle, and the results (zoomed-in) of: (g) GVF, (h) BVF, (i) MAC (j) FVF; (k) an image with an initial contour overlapping the eye, and the results (zoomed-in) of (l) GVF, (m) BVF, (n) MAC (o) FVF.



(a) (b)

Fig. 12. A visual inspection of the FVF generated contour: The images in (a) and (b) show the FVF detected contour (blue) overlaid with the ground truth (red).

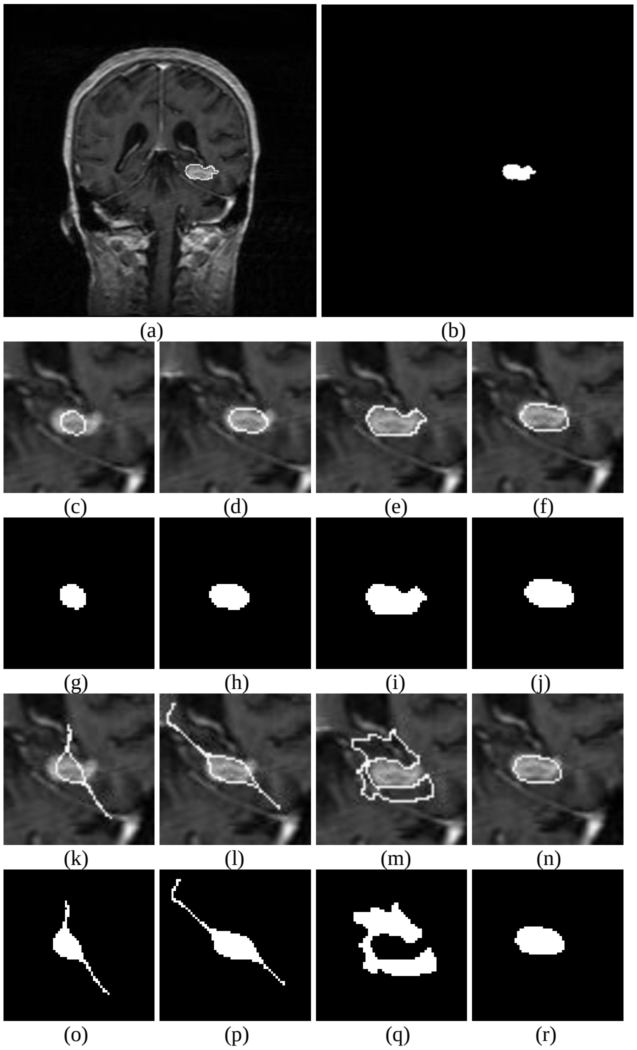

(a) (b)
(c) (d) (e) (f)
(g) (h) (i) (j)
(k) (l) (m) (n)
(o) (p) (q) (r)

Fig. 13. (a) Ground-truth and (b) the segmented region of ground-truth of Image #4 in Table 1, and the results of (c) GVF, (d) BVF, (e) MAC, (f) FVF, and the segmented regions of (g) GVF, (h) BVF, (i) MAC, (j) FVF, when the initial contour (not shown) is inside the brain tumor; and the results of (k) GVF, (l) BVF, (m) MAC, (n) FVF, and the segmented regions of (o) GVF, (p) BVF, (q) MAC, (r) FVF, when the initial contour (not shown) is inside the brain tumor.

Table I. Quantitative analysis of GVF, BVF, MAC and FVF based on IBSR brain tumor MRI images.

Image	Method	TM (inside)	TM (outside)	TM (overlap)
#1	GVF	0.173	0.340	0.201
	BVF	0.004	0.350	0.430
	MAC	0.013	0.061	0.013
	FVF	**0.515**	**0.519**	**0.521**
#2	GVF	0.005	0.217	0.005
	BVF	0.014	0.468	**0.664**
	MAC	0.015	0.048	0.022
	FVF	**0.582**	**0.589**	0.598
#3	GVF	0.004	0.117	0.004
	BVF	0.004	0.572	0.007
	MAC	**0.722**	0.074	**0.719**
	FVF	0.709	**0.716**	0.710
#4	GVF	0.410	0.418	0.414
	BVF	0.640	0.574	0.649
	MAC	**0.876**	0.080	**0.842**
	FVF	0.667	**0.656**	0.661
#5	GVF	0.005	0.092	0.009
	BVF	0.585	0.478	0.576
	MAC	0.005	0.075	0.041
	FVF	**0.653**	**0.653**	**0.658**
#6	GVF	0.005	0.189	0.010
	BVF	0.250	0.372	0.275
	MAC	**0.791**	0.056	0.060
	FVF	0.561	**0.563**	**0.571**
#7	GVF	0.294	0.375	0.317
	BVF	0.540	0.520	0.531
	MAC	**0.764**	0.066	0.036
	FVF	0.565	**0.571**	**0.588**
#8	GVF	0.415	0.370	0.398
	BVF	0.591	0.526	0.572
	MAC	0.012	0.072	0.056
	FVF	**0.608**	**0.607**	**0.599**
#9	GVF	0.000	0.300	0.000
	BVF	0.361	0.000	0.631
	MAC	**0.787**	0.079	**0.781**
	FVF	0.599	**0.597**	0.598
#10	GVF	0.004	0.106	0.011
	BVF	0.000	**0.626**	0.386
	MAC	**0.813**	0.162	0.041
	FVF	0.617	0.616	**0.589**

Table II. Mean, median and standard deviation of the TM of GVF, BVF, MAC and FVF.

Method	Mean	Median	Std
GVF	0.17	0.15	0.17
BVF	0.41	0.50	0.23
MAC	0.27	0.07	0.35
FVF	0.61	0.60	0.05

Table III. Comparison between FVF and GVF, BVF, and MAC using mixed effects model.

Comparison	Estimate	Standard Error	P-value
FVF-GVF	0.4349	0.07084	<0.0001
FVF-BVF	0.2020	0.07084	0.0072
FVF-MAC	0.3358	0.07084	<0.0001