Snakes, Shapes, and Gradient Vector Flow

Chenyang Xu, *Student Member, IEEE,* and Jerry L. Prince, *Senior Member, IEEE*

***Abstract*— Snakes, or active contours, are used extensively in computer vision and image processing applications, particularly to locate object boundaries. Problems associated with initialization and poor convergence to boundary concavities, however, have limited their utility. This paper presents a new external force for active contours, largely solving both problems. This external force, which we call *gradient vector flow* (GVF), is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image. It differs fundamentally from traditional snake external forces in that it cannot be written as the negative gradient of a potential function, and the corresponding snake is formulated directly from a force balance condition rather than a variational formulation. Using several two-dimensional (2-D) examples and one three-dimensional (3-D) example, we show that GVF has a large capture range and is able to move snakes into boundary concavities.**

***Index Terms*—Active contour models, deformable surface models, edge detection, gradient vector flow, image segmentation, shape representation and recovery, snakes.**

I. Introduction

SNAKES [1], or *active contours*, are curves defined within an image domain that can move under the influence of internal forces coming from within the curve itself and external forces computed from the image data. The internal and external forces are defined so that the snake will conform to an object boundary or other desired features within an image. Snakes are widely used in many applications, including edge detection [1], shape modeling [2], [3], segmentation [4], [5], and motion tracking [4], [6].

There are two general types of active contour models in the literature today: *parametric active contours* [1] and *geometric active contours* [7]–[9]. In this paper, we focus on parametric active contours, although we expect our results to have applications in geometric active contours as well. Parametric active contours synthesize parametric curves within an image domain and allow them to move toward desired features, usually edges. Typically, the curves are drawn toward the edges by *potential forces*, which are defined to be the negative gradient of a potential function. Additional forces, such as pressure forces [10], together with the potential forces comprise the *external forces*. There are also *internal forces* designed to hold the curve together (elasticity forces) and to keep it from bending too much (bending forces).

There are two key difficulties with parametric active contour algorithms. First, the initial contour must, in general, be close to the true boundary or else it will likely converge to the wrong result. Several methods have been proposed to address this problem including multiresolution methods [11], pressure forces [10], and distance potentials [12]. The basic idea is to increase the *capture range* of the external force fields and to guide the contour toward the desired boundary. The second problem is that active contours have difficulties progressing into *boundary concavities* [13], [14]. There is no satisfactory solution to this problem, although pressure forces [10], control points [13], domain-adaptivity [15], directional attractions [14], and the use of solenoidal fields [16] have been proposed. However, most of the methods proposed to address these problems solve only one problem while creating new difficulties. For example, multiresolution methods have addressed the issue of capture range, but specifying how the snake should move across different resolutions remains problematic. Another example is that of pressure forces, which can push an active contour into boundary concavities, but cannot be too strong or “weak” edges will be overwhelmed [17]. Pressure forces must also be initialized to push out or push in, a condition that mandates careful initialization.

In this paper, we present a new class of external forces for active contour models that addresses both problems listed above. These fields, which we call *gradient vector flow* (GVF) fields, are dense vector fields derived from images by minimizing a certain energy functional in a variational framework. The minimization is achieved by solving a pair of decoupled linear partial differential equations that diffuses the gradient vectors of a gray-level or binary edge map computed from the image. We call the active contour that uses the GVF field as its external force a *GVF snake*. The GVF snake is distinguished from nearly all previous snake formulations in that its external forces cannot be written as the negative gradient of a potential function. Because of this, it cannot be formulated using the standard energy minimization framework; instead, it is specified directly from a force balance condition.

Particular advantages of the GVF snake over a traditional snake are its insensitivity to initialization and its ability to move into boundary concavities. As we show in this paper, its initializations can be inside, outside, or across the object’s boundary. Unlike pressure forces, the GVF snake does not need prior knowledge about whether to shrink or expand toward the boundary. The GVF snake also has a large capture range, which means that, barring interference from other objects, it can be initialized far away from the boundary. This increased capture range is achieved through a diffusion process that does not blur the edges themselves, so multiresolution

Manuscript received November 1, 1996; revised March 17, 1997. This work was supported by NSF Presidential Faculty Fellow Award MIP93-50336. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Guillermo Sapiro.

The authors are with the Image Analysis and Communications Laboratory, Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: prince@jhu.edu).

Publisher Item Identifier S 1057-7149(98)01745-X.

methods are not needed. The external force model that is closest in spirit to GVF is the distance potential forces of Cohen and Cohen [12]. Like GVF, these forces originate from an edge map of the image and can provide a large capture range. We show, however, that unlike GVF, distance potential forces cannot move a snake into boundary concavities. We believe that this is a property of all conservative forces that characterize nearly all snake external forces, and that exploring nonconservative external forces, such as GVF, is an important direction for future research in active contour models.

We note that part of the work reported in this paper has appeared in the conference paper [18].

II. Background

A. Parametric Snake Model

A *traditional snake* is a curve $\mathbf{x}(s) = [x(s), y(s)]$, $s \in [0, 1]$, that moves through the spatial domain of an image to minimize the energy functional

$$E = \int_0^1 \frac{1}{2}[\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2] + E_{\text{ext}}(\mathbf{x}(s))\, ds \tag{1}$$

where α and β are weighting parameters that control the snake's tension and rigidity, respectively, and $\mathbf{x}'(s)$ and $\mathbf{x}''(s)$ denote the first and second derivatives of $\mathbf{x}(s)$ with respect to s. The external energy function E_{ext} is derived from the image so that it takes on its smaller values at the features of interest, such as boundaries. Given a gray-level image $I(x, y)$, viewed as a function of continuous position variables (x, y), typical external energies designed to lead an active contour toward step edges [1] are

$$E_{\text{ext}}^{(1)}(x, y) = -|\nabla I(x, y)|^2 \tag{2}$$

$$E_{\text{ext}}^{(2)}(x, y) = -|\nabla[G_\sigma(x, y) * I(x, y)]|^2 \tag{3}$$

where $G_\sigma(x, y)$ is a two-dimensional Gaussian function with standard deviation σ and ∇ is the gradient operator. If the image is a line drawing (black on white), then appropriate external energies include [10]:

$$E_{\text{ext}}^{(3)}(x, y) = I(x, y) \tag{4}$$

$$E_{\text{ext}}^{(4)}(x, y) = G_\sigma(x, y) * I(x, y). \tag{5}$$

It is easy to see from these definitions that larger σ's will cause the boundaries to become blurry. Such large σ's are often necessary, however, in order to increase the capture range of the active contour.

A snake that minimizes E must satisfy the Euler equation

$$\alpha\mathbf{x}''(s) - \beta\mathbf{x}''''(s) - \nabla E_{\text{ext}} = 0. \tag{6}$$

This can be viewed as a force balance equation

$$\mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}}^{(\text{p})} = 0 \tag{7}$$

where $\mathbf{F}_{\text{int}} = \alpha\mathbf{x}''(s) - \beta\mathbf{x}''''(s)$ and $\mathbf{F}_{\text{ext}}^{(\text{p})} = -\nabla E_{\text{ext}}$. The internal force $\mathbf{F}_{\text{int}}$ discourages stretching and bending while the external potential force $\mathbf{F}_{\text{ext}}^{(\text{p})}$ pulls the snake toward the desired image edges.

To find a solution to (6), the snake is made dynamic by treating $\mathbf{x}$ as function of time t as well as s—i.e., $\mathbf{x}(s, t)$. Then, the partial derivative of $\mathbf{x}$ with respect to t is then set equal to the left hand side of (6) as follows:

$$\mathbf{x}_t(s, t) = \alpha\mathbf{x}''(s, t) - \beta\mathbf{x}''''(s, t) - \nabla E_{\text{ext}}. \tag{8}$$

When the solution $\mathbf{x}(s, t)$ stabilizes, the term $\mathbf{x}_t(s, t)$ vanishes and we achieve a solution of (6). A numerical solution to (8) can be found by discretizing the equation and solving the discrete system iteratively (cf., [1]). We note that most snake implementations use either a parameter which multiplies $\mathbf{x}_t$ in order to control the temporal step-size, or a parameter to multiply ∇E_{ext}, which permits separate control of the external force strength. In this paper, we normalize the external forces so that the maximum magnitude is equal to one, and use a unit temporal step-size for all the experiments.

B. Behavior of Traditional Snakes

An example of the behavior of a traditional snake is shown in Fig. 1. Fig. 1(a) shows a 64 × 64-pixel line-drawing of a U-shaped object (shown in gray) having a boundary concavity at the top. It also shows a sequence of curves (in black) depicting the iterative progression of a traditional snake ($\alpha = 0.6$, $\beta = 0.0$) initialized outside the object but within the capture range of the potential force field. The potential force field $\mathbf{F}_{\text{ext}}^{(\text{p})} = -\nabla E_{\text{ext}}^{(4)}$ where $\sigma = 1.0$ pixel is shown in Fig. 1(b). We note that the final solution in Fig. 1(a) solves the Euler equations of the snake formulation, but remains split across the concave region.

The reason for the poor convergence of this snake is revealed in Fig. 1(c), where a close-up of the external force field within the boundary concavity is shown. Although the external forces correctly point toward the object boundary, within the boundary concavity the forces point horizontally *in opposite directions*. Therefore, the active contour is pulled apart toward each of the "fingers" of the U-shape, but not made to progress downward into the concavity. There is no choice of α and β that will correct this problem.

Another key problem with traditional snake formulations, the problem of limited capture range, can be understood by examining Fig. 1(b). In this figure, we see that the magnitude of the external forces die out quite rapidly away from the object boundary. Increasing σ in (5) will increase this range, but the boundary localization will become less accurate and distinct, ultimately obliterating the concavity itself when σ becomes too large.

Cohen and Cohen [12] proposed an external force model that significantly increases the capture range of a traditional snake. These external forces are the negative gradient of a potential function that is computed using a Euclidean (or chamfer) distance map. We refer to these forces as *distance potential forces* to distinguish them from the traditional potential forces defined in Section II-A. Fig. 2 shows the performance of a snake using distance potential forces. Fig. 2(a) shows both the U-shaped object (in gray) and a sequence of contours (in black) depicting the progression of the snake from its initialization far from the object to its final configuration. The

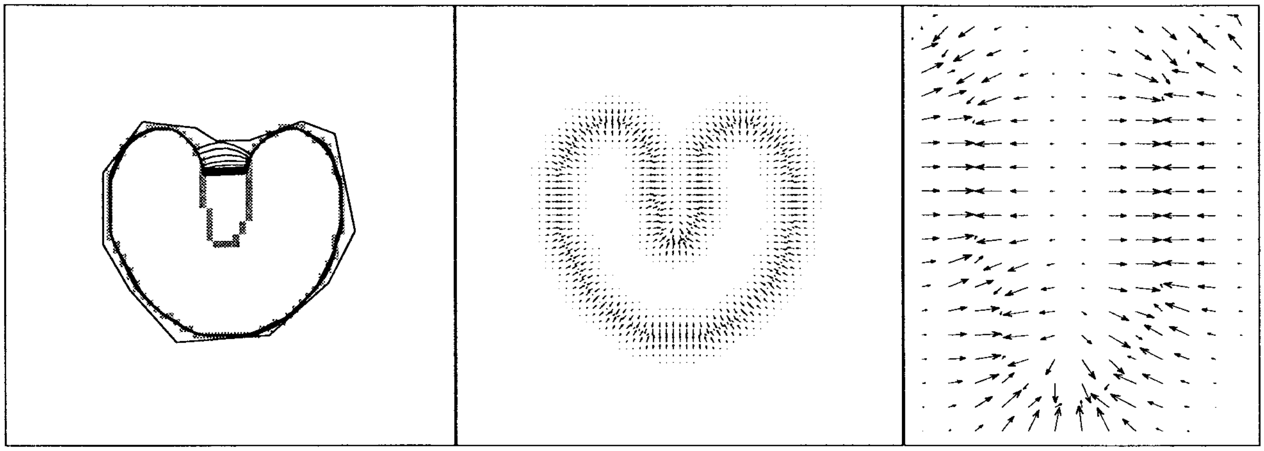


Fig. 1. (a) Convergence of a snake using (b) traditional potential forces, and (c) shown close-up within the boundary concavity.

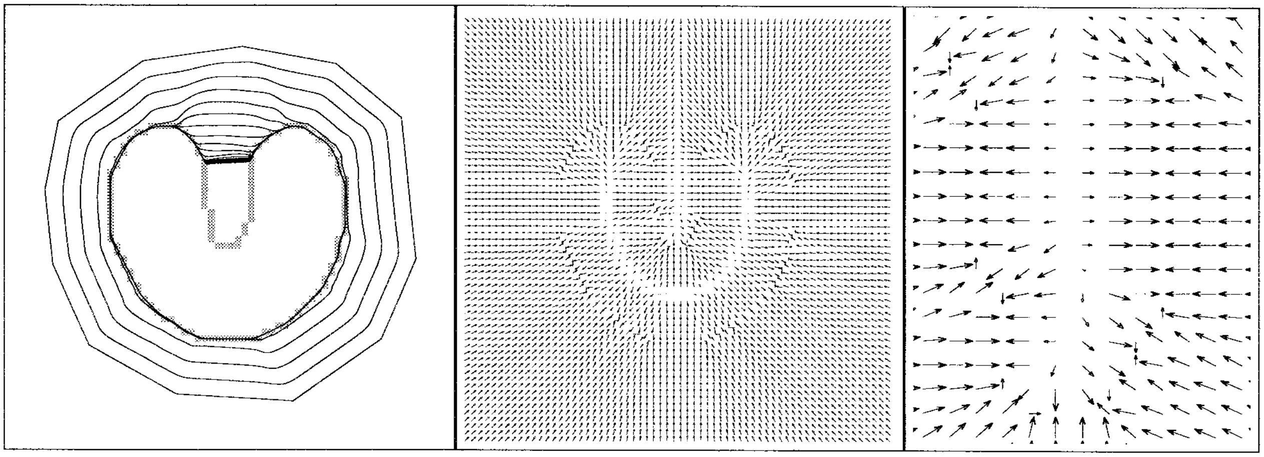


Fig. 2. (a) Convergence of a snake using (b) distance potential forces, and (c) shown close-up within the boundary concavity.

distance potential forces shown in Fig. 2(b) have vectors with large magnitudes far away from the object, explaining why the capture range is large for this external force model.

As shown in Fig. 2(a), this snake also fails to converge to the boundary concavity. This can be explained by inspecting the magnified portion of the distance potential forces shown in Fig. 2(c). We see that, like traditional potential forces, these forces also point horizontally in opposite directions, which pulls the snake apart but not downward into the boundary concavity. We note that Cohen and Cohen's modification to the basic distance potential forces, which applies a nonlinear transformation to the distance map [12], does not change the direction of the forces, only their magnitudes. Therefore, the problem of convergence to boundary concavities is not solved by distance potential forces.

C. Generalized Force Balance Equations

The snake solutions shown in Figs. 1(a) and 2(a) both satisfy the Euler equations (6) for their respective energy model. Accordingly, the poor final configurations can be attributed to convergence to a local minimum of the objective function (1). Several researchers have sought solutions to this problem by formulating snakes directly from a force balance equation in which the standard external force $\mathbf{F}_{\text{ext}}^{(\text{p})}$ is replaced by a more general external force $\mathbf{F}_{\text{ext}}^{(\text{g})}$ as follows:

$$\mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}}^{(\text{g})} = 0. \tag{9}$$

The choice of $\mathbf{F}_{\text{ext}}^{(\text{g})}$ can have a profound impact on both the implementation and the behavior of a snake. Broadly speaking, the external forces $\mathbf{F}_{\text{ext}}^{(\text{g})}$ can be divided into two classes: static and dynamic. Static forces are those that are computed from the image data, and do not change as the snake progresses. Standard snake potential forces are static external forces. Dynamic forces are those that change as the snake deforms.

Several types of dynamic external forces have been invented to try to improve upon the standard snake potential forces. For example, the forces used in multiresolution snakes [11] and the pressure forces used in balloons [10] are dynamic external forces. The use of multiresolution schemes and pressure forces, however, adds complexity to a snake's implementation and unpredictability to its performance. For example, pressure forces must be initialized to either push out or push in, and may overwhelm weak boundaries if they act too strongly [17]. Conversely, they may not move into boundary concavities if they are pushing in the wrong direction or act too weakly.

In this paper, we present a new type of *static external force*, one that does not change with time or depend on the position of the snake itself. The underlying mathematical premise for this new force comes from the Helmholtz theorem (cf., [19]), which states that the most general static vector field can be decomposed into two components: an irrotational (curl-free) component and a solenoidal (divergence-free) component.[1] An external potential force generated from the variational formulation of a traditional snake must enter the force balance equation (6) as a static irrotational field, since it is the gradient of a potential function. Therefore, a more general static field $\mathbf{F}_{\text{ext}}^{(\text{g})}$ can be obtained by allowing the possibility that it comprises *both* an irrotational component and a solenoidal component. Our previous paper [16] explored the idea of constructing a separate solenoidal field from an image, which was then added to a standard irrotational field. In the following section, we pursue a more natural approach in which the external force field is designed to have the desired properties of both a large capture range and the presence of forces that point into boundary concavities. The resulting formulation produces external force fields that can be expected to have both irrotational and solenoidal components.

III. Gradient Vector Flow Snake

Our overall approach is to use the force balance condition (7) as a starting point for designing a snake. We define below a new static external force field $\mathbf{F}_{\text{ext}}^{(\text{g})} = \mathbf{v}(x, y)$, which we call the *gradient vector flow* (GVF) field. To obtain the corresponding dynamic snake equation, we replace the potential force $-\nabla E_{\text{ext}}$ in (8) with $\mathbf{v}(x, y)$, yielding

$$\mathbf{x}_t(s, t) = \alpha \mathbf{x}''(s, t) - \beta \mathbf{x}''''(s, t) + \mathbf{v}. \tag{10}$$

We call the parametric curve solving the above dynamic equation a *GVF snake*. It is solved numerically by discretization and iteration, in identical fashion to the traditional snake.

Although the final configuration of a GVF snake will satisfy the force-balance equation (7), this equation does not, in general, represent the Euler equations of the energy minimization problem in (1). This is because $\mathbf{v}(x, y)$ will not, in general, be an irrotational field. The loss of this optimality property, however, is well-compensated by the significantly improved performance of the GVF snake.

A. Edge Map

We begin by defining an *edge map* $f(x, y)$ derived from the image $I(x, y)$ having the property that it is larger near the image edges.[2] We can use any gray-level or binary edge map defined in the image processing literature (cf., [20]); for example, we could use

$$f(x, y) = -E_{\text{ext}}^{(i)}(x, y) \tag{11}$$

where $i = 1$, 2, 3, or 4. Three general properties of edge maps are important in the present context. First, the gradient of an edge map ∇f has vectors pointing toward the edges, which are normal to the edges at the edges. Second, these vectors generally have large magnitudes only in the immediate vicinity of the edges. Third, in homogeneous regions, where $I(x, y)$ is nearly constant, ∇f is nearly zero.

Now consider how these properties affect the behavior of a traditional snake when the gradient of an edge map is used as an external force. Because of the first property, a snake initialized close to the edge will converge to a stable configuration near the edge. This is a highly desirable property. Because of the second property, however, the capture range will be very small, in general. Because of the third property, homogeneous regions will have no external forces whatsoever. These last two properties are undesirable. Our approach is to keep the highly desirable property of the gradients near the edges, but to extend the gradient map farther away from the 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.

B. Gradient Vector Flow

We define the gradient vector flow field to be the vector field $\mathbf{v}(x, y) = [u(x, y), v(x, y)]$ that minimizes the energy functional

$$\mathcal{E} = \iint \mu({u_x}^2 + {u_y}^2 + {v_x}^2 + {v_y}^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 \, dxdy. \tag{12}$$

This variational formulation follows a standard principle, that of making the result smooth when there is no data. In particular, we see that 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 $\mathbf{v} = \nabla f$. This produces the desired effect of keeping $\mathbf{v}$ 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 (more noise, increase μ).

We note that the smoothing term—the first term within the integrand of (12)—is the same term used by Horn and Schunck in their classical formulation of optical flow [21]. It has recently been shown that this term corresponds to an equal penalty on the divergence and curl of the vector field [22]. Therefore, the vector field resulting from this minimization can be expected to be neither entirely irrotational nor entirely solenoidal.

[1] Irrotational fields are sometimes called conservative fields; they can be represented as the gradient of a scalar potential function.

[2] Other features can be sought by redefining $f(x, y)$ to be larger at the desired features.

Using the *calculus of variations* [23], it can be shown that the GVF field can be found by solving the following Euler equations

$$\mu\nabla^2 u - (u - f_x)({f_x}^2 + {f_y}^2) = 0 \tag{13a}$$

$$\mu\nabla^2 v - (v - f_y)({f_x}^2 + {f_y}^2) = 0 \tag{13b}$$

where ∇^2 is the Laplacian operator. These equations provide further intuition behind the GVF formulation. We note that in a homogeneous region [where $I(x, y)$ is constant], the second term in each equation is zero because the gradient of $f(x, y)$ is zero. Therefore, within such a region, u and v are each determined by Laplace's equation, and the resulting GVF field is interpolated from the region's boundary, reflecting a kind of competition among the boundary vectors. This explains why GVF yields vectors that point into boundary concavities.

C. Numerical Implementation

Equations (13a) and (13b) can be solved by treating u and v as functions of time and solving

$$\begin{aligned} u_t(x,y,t) &= \mu\nabla^2 u(x,y,t) - [u(x,y,t) - f_x(x,y)] \\ &\quad \cdot [f_x(x,y)^2 + f_y(x,y)^2] \end{aligned} \tag{14a}$$

$$\begin{aligned} v_t(x,y,t) &= \mu\nabla^2 v(x,y,t) - [v(x,y,t) - f_y(x,y)] \\ &\quad \cdot [f_x(x,y)^2 + f_y(x,y)^2]. \end{aligned} \tag{14b}$$

The steady-state solution of these linear parabolic equations is the desired solution of the Euler equations (13a) and (13b). Note that these equations are decoupled, and therefore can be solved as separate scalar partial differential equations in u and v. The equations in (14) are known as *generalized diffusion equations*, and are known to arise in such diverse fields as heat conduction, reactor physics, and fluid flow [24]. Here, they have appeared from our description of desirable properties of snake external force fields as represented in the energy functional of (12).

For convenience, we rewrite (14) as follows:

$$u_t(x,y,t) = \mu\nabla^2 u(x,y,t) - b(x,y)u(x,y,t) + c^1(x,y) \tag{15a}$$

$$v_t(x,y,t) = \mu\nabla^2 v(x,y,t) - b(x,y)v(x,y,t) + c^2(x,y) \tag{15b}$$

where

$$\begin{aligned} b(x, y) &= f_x(x, y)^2 + f_y(x, y)^2 \\ c^1(x, y) &= b(x, y)f_x(x, y) \\ c^2(x, y) &= b(x, y)f_y(x, y). \end{aligned}$$

Any digital image gradient operator (cf., [20]) can be used to calculate f_x and f_y. In the examples shown in this paper, we use simple central differences. The coefficients $b(x, y)$, $c^1(x, y)$, and $c^2(x, y)$, can then be computed and fixed for the entire iterative process.

To set up the iterative solution, let the indices i, j, and n correspond to x, y, and t, respectively, and let the spacing between pixels be Δx and Δy and the time step for each iteration be Δt. Then the required partial derivatives can be approximated as

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

Substituting these approximations into (15) gives our iterative solution to GVF as follows:

$$\begin{aligned} u_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)u_{i,j}^{n} + r(u_{i+1,j}^{n} + u_{i,j+1}^{n} + u_{i-1,j}^{n} \\ &\quad + u_{i,j-1}^{n} - 4u_{i,j}^{n}) + c_{i,j}^{1}\Delta t \end{aligned} \tag{16a}$$

$$\begin{aligned} v_{i,j}^{n+1} &= (1 - b_{i,j}\Delta t)v_{i,j}^{n} + r(v_{i+1,j}^{n} + v_{i,j+1}^{n} + v_{i-1,j}^{n} \\ &\quad + v_{i,j-1}^{n} - 4v_{i,j}^{n}) + c_{i,j}^{2}\Delta t \end{aligned} \tag{16b}$$

where

$$r = \frac{\mu \Delta t}{\Delta x \Delta y}. \tag{17}$$

Convergence of the above iterative process is guaranteed by a standard result in the theory of numerical methods (cf., [25]). Provided that b, c^1, and c^2 are bounded, (16) is stable whenever the Courant–Friedrichs–Lewy step-size restriction $r \leq 1/4$ is maintained. Since normally Δx, Δy, and μ are fixed, using the definition of r in (17), we find that the following restriction on the time-step Δt must be maintained in order to guarantee convergence of GVF:

$$\Delta t \leq \frac{\Delta x \Delta y}{4\mu}. \tag{18}$$

The intuition behind this condition is revealing. First, convergence can be made to be faster on coarser images—i.e., when Δx and Δy are larger. Second, when μ is large and the GVF is expected to be a smoother field, the convergence rate will be slower (since Δt must be kept small).

Our 2-D GVF computations were implemented using MATLAB[3] code. For an $N = 256 \times 256$-pixel image on an SGI Indigo-2, typical computation times are 8 s for the traditional potential forces (written in C), 155 s for the distance potential forces (Euclidean distance map, written in MATLAB), and 420 s for the GVF forces (written in MATLAB, using $\sqrt{N}$ iterations). The computation time of GVF can be substantially reduced by using optimized code in C or FORTRAN. For example, we have implemented 3-D GVF (see Section V-B) in C, and computed GVF with 150 iterations on a 256 × 256 × 60-voxel image in 31 min. Accounting for the size difference and extra dimension, we conclude that written in C, GVF computation for a 2-D 256 × 256-pixel image would take approximately 53 s. Algorithm optimization such as use of the multigrid method should yield further improvements.

[3] Mathworks, Natick, MA

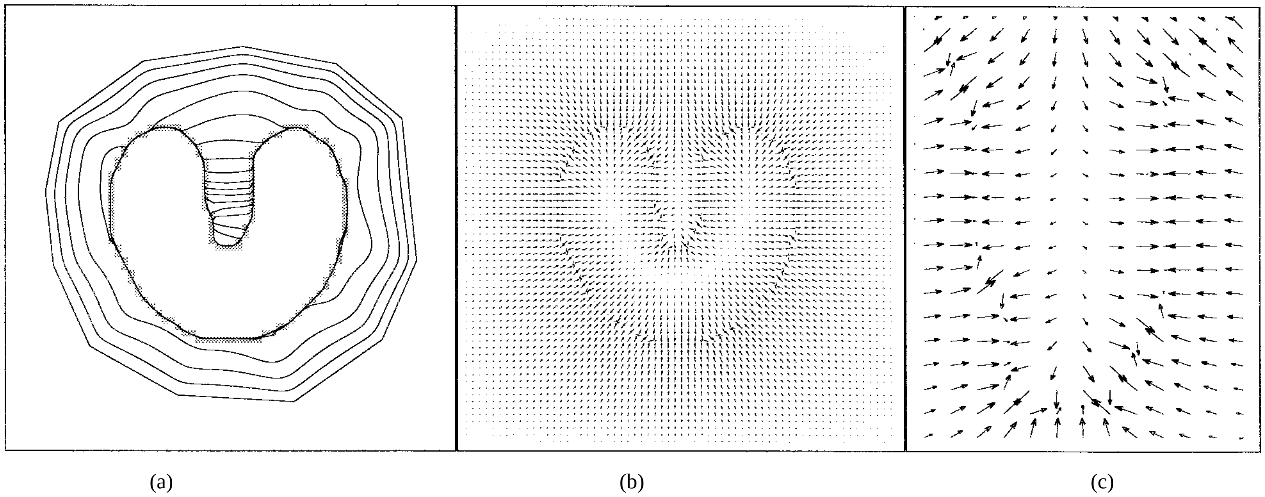


Fig. 3. (a) Convergence of a snake using (b) GVF external forces, and (c) shown close-up within the boundary concavity.

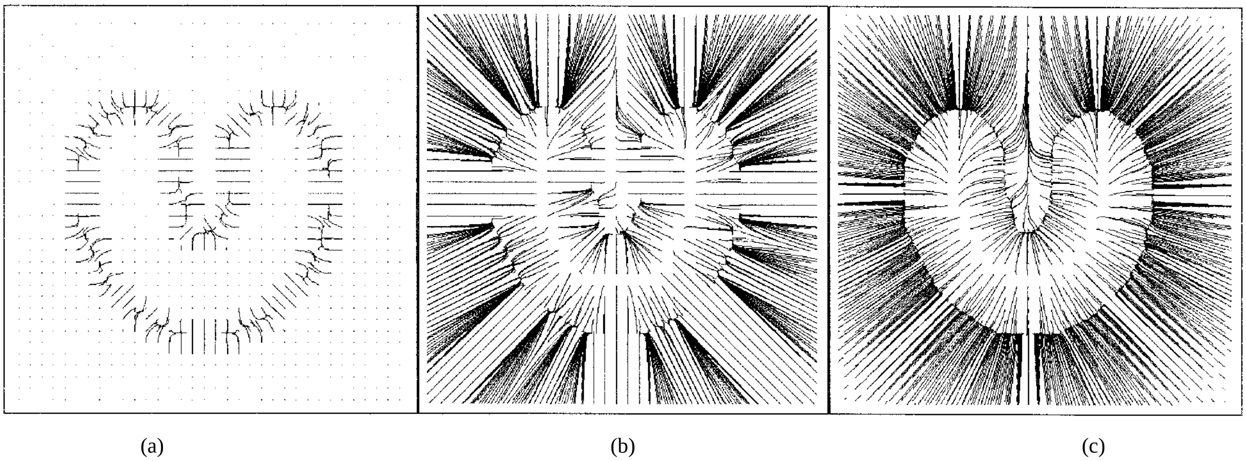


Fig. 4. Streamlines originating from an array of 32 × 32 particles in (a) a traditional potential force field, (b) a distance potential force field, and (c) a GVF force field.

IV. GVF Fields and Snakes: Demonstrations

This section shows several examples of GVF field computations on simple objects and demonstrates several key properties of GVF snakes. We used $\alpha = 0.6$ and $\beta = 0.0$ for all snakes and $\mu = 0.2$ for GVF. The snakes were dynamically reparameterized to maintain contour point separation to within 0.5–1.5 pixels (cf., [26]). All edge maps used in GVF computations were normalized to the range [0, 1].

A. Convergence to Boundary Concavity

In our first experiment, we computed the GVF field for the same U-shaped object used in Figs. 1 and 2. The results are shown in Fig. 3. Comparing the GVF field, shown in Fig. 3(b), to the traditional potential force field of Fig. 1(b), reveals several key differences. First, like the distance potential force field [Fig. 2(b)], the GVF field has a much larger capture range than traditional potential forces. A second observation, which can be seen in the close-up of Fig. 3(c), is that the GVF vectors within the boundary concavity at the top of the U-shape have a downward component. This stands in stark contrast to both the traditional potential forces of Fig. 1(c) and the distance potential forces of Fig. 2(c). Finally, it can be seen from Fig. 3(b) that the GVF field behaves in an analogous fashion when viewed from the inside of the object. In particular, the GVF vectors are pointing upward into the "fingers" of the U-shape, which represent concavities from this perspective.

Fig. 3(a) shows the initialization, progression, and final configuration of a GVF snake. The initialization is the same as that of Fig. 2(a), and the snake parameters are the same as those in Figs. 1 and 2. Clearly, the GVF snake has a broad capture range and superior convergence properties. The final snake configuration closely approximates the true boundary, arriving at a subpixel interpolation through bilinear interpolation of the GVF force field.

B. Streamlines

Streamlines are the paths over which free particles move when placed in an external force field. By looking at their streamlines, we can examine the capture ranges and motion inducing properties for various snake external forces. Fig. 4 shows the streamlines of points arranged on a 32 × 32 grid

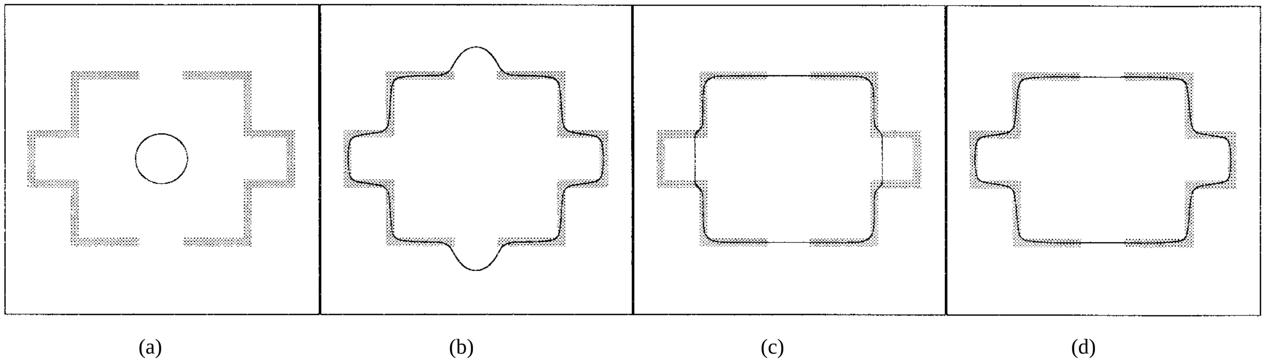


Fig. 5. (a) Initial curve and snake results from (b) a balloon with an outward pressure, (c) a distance potential force snake, and (d) a GVF snake.

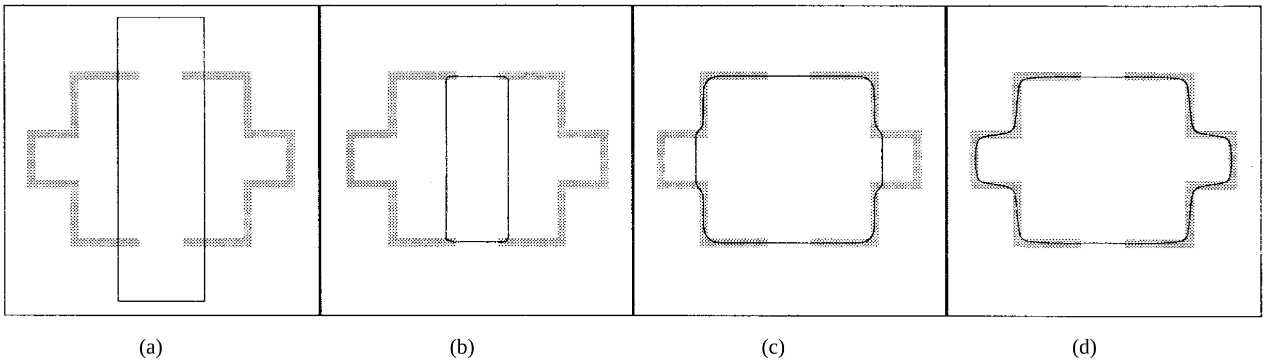


Fig. 6. (a) Initial curve and snake results from (b) a traditional snake, (c) a distance potential force snake, and (d) a GVF snake.

for the traditional potential forces, distance potential forces, and GVF forces used in the simulations of Figs. 1–3.

Several properties can be observed from these figures. First, the capture ranges of the GVF force field and the distance potential force field are clearly much larger than that of the traditional potential force field. In fact, both distance potential forces and GVF forces will attract a snake that is initialized on the image border. Second, it is clear that GVF is the only force providing both a downward force within the boundary concavity at the top of the U-shape and an upward force within the "fingers" of the U-shape. In contrast, both traditional snake forces and distance potential forces provide only sideways forces in these regions. Third, the distance potential forces appear to have boundary points that act as regional points of attraction. In contrast, the GVF forces attract points uniformly toward the boundary.

C. Snake Initialization and Convergence

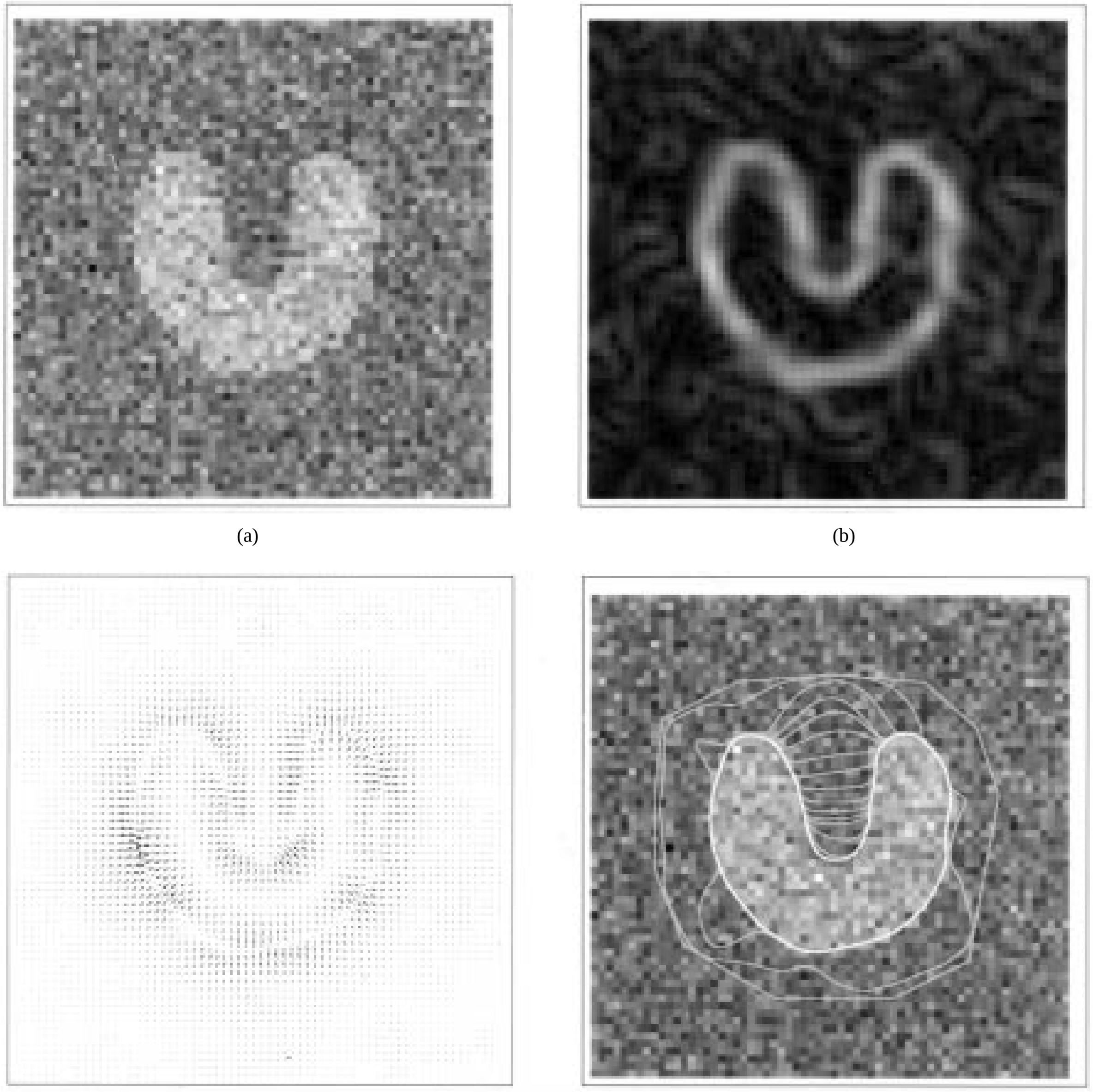
In this section, we present several examples that compare different snake models with the GVF snake, showing various effects related to initialization, boundary concavities, and subjective contours. The object under study is the line drawing drawn in gray in both Figs. 5 and 6. This figure may depict, for example, the boundary of a room having two doors at the top and bottom and two alcoves at the left and right. The open doors at the top and bottom represent subjective contours that we desire to connect using the snake (cf., [1]).

The snake results shown in Fig. 5(b)–5(d) all used the initialization shown in Fig. 5(a). We first note that for this initialization, the traditional potential forces were too weak to overpower the snake's internal forces, and the snake shrank to a point at the center of the figure (result not shown). To try to fix this problem, a balloon model with outward pressure forces just strong enough to cause the snake to expand into the boundary concavities was implemented; this result is shown in Fig. 5(b). Clearly, the pressure forces also caused the balloon to bulge outward through the openings at the top and bottom, and therefore the subjective contours are not reconstructed well.

The snake result obtained using the distance potential force model is shown in Fig. 5(c). Clearly, the capture range is now adequate and the subjective boundaries at the top and bottom are reconstructed well. But this snake fails to find the boundary concavities at the left and right, for the same reason that it could not proceed into the top of the U-shaped object of the previous sections. The GVF snake result, shown in Fig. 5(d), is clearly the best result. It has reconstructed both the subjective boundaries and the boundary concavities quite well. The slight rounding of corners, which can also be seen in Figs. 5(b) and 5(c), is a fundamental characteristic of snakes caused by the regularization coefficients α and β.[4]

The snake results shown in Fig. 6(b)–6(d) all used the initialization shown in Fig. 6(a), which is deliberately placed across the boundary. In this case, the balloon model cannot be sensibly applied because it is not clear whether to apply inward or outward pressure forces. Instead, the result of a snake with traditional potential forces is shown in Fig. 6(b).

[4] The effect is only caused by α in this example since $\beta = 0$.



(a) (b)

(c) (d)

Fig. 7. (a) Noisy 64 × 64-pixel image of a U-shaped object; (b) edge map $|\nabla(G_\sigma * I)|^2$ with $\sigma = 1.5$; (c) GVF external force field; and (d) convergence of the GVF snake.

This snake stops at a very undesirable configuration because its only points of contact with the boundary are normal to it and the remainder of the snake is outside the capture range of the other parts of the boundary. The snake resulting from distance potential forces is shown in Fig. 6(c). This result shows that although the distance potential force snake possesses an insensitivity to initialization, it is incapable of progressing into boundary concavities. The GVF snake result, shown in Fig. 6(d), is again the best result. The GVF snake appears to have both an insensitivity to initialization and an ability to progress into boundary concavities.

V. Gray-Level Images and Higher Dimensions

In this section, we describe and demonstrate how GVF can be used in gray-level imagery and in higher dimensions.

A. Gray-Level Images

The underlying formulation of GVF is valid for gray-level images as well as binary images. To compute GVF for gray-level images, the edge-map function $f(x, y)$ must first be calculated. Two possibilities are $f^{(1)}(x, y) = |\nabla I(x, y)|$ or $f^{(2)}(x, y) = |\nabla[G_\sigma(x, y) * I(x, y)]|$, where the latter is more robust in the presence of noise. Other more complicated noise-removal techniques such as median filtering, morphological filtering, and anisotropic diffusion could also be used to improve the underlying edge map. Given an edge-map function and an approximation to its gradient, GVF is computed in the usual way using (16).

Fig. 7(a) shows a gray-level image of the U-shaped object corrupted by additive white Gaussian noise; the signal-to-noise ratio is 6 dB. Fig. 7(b) shows an edge-map computed

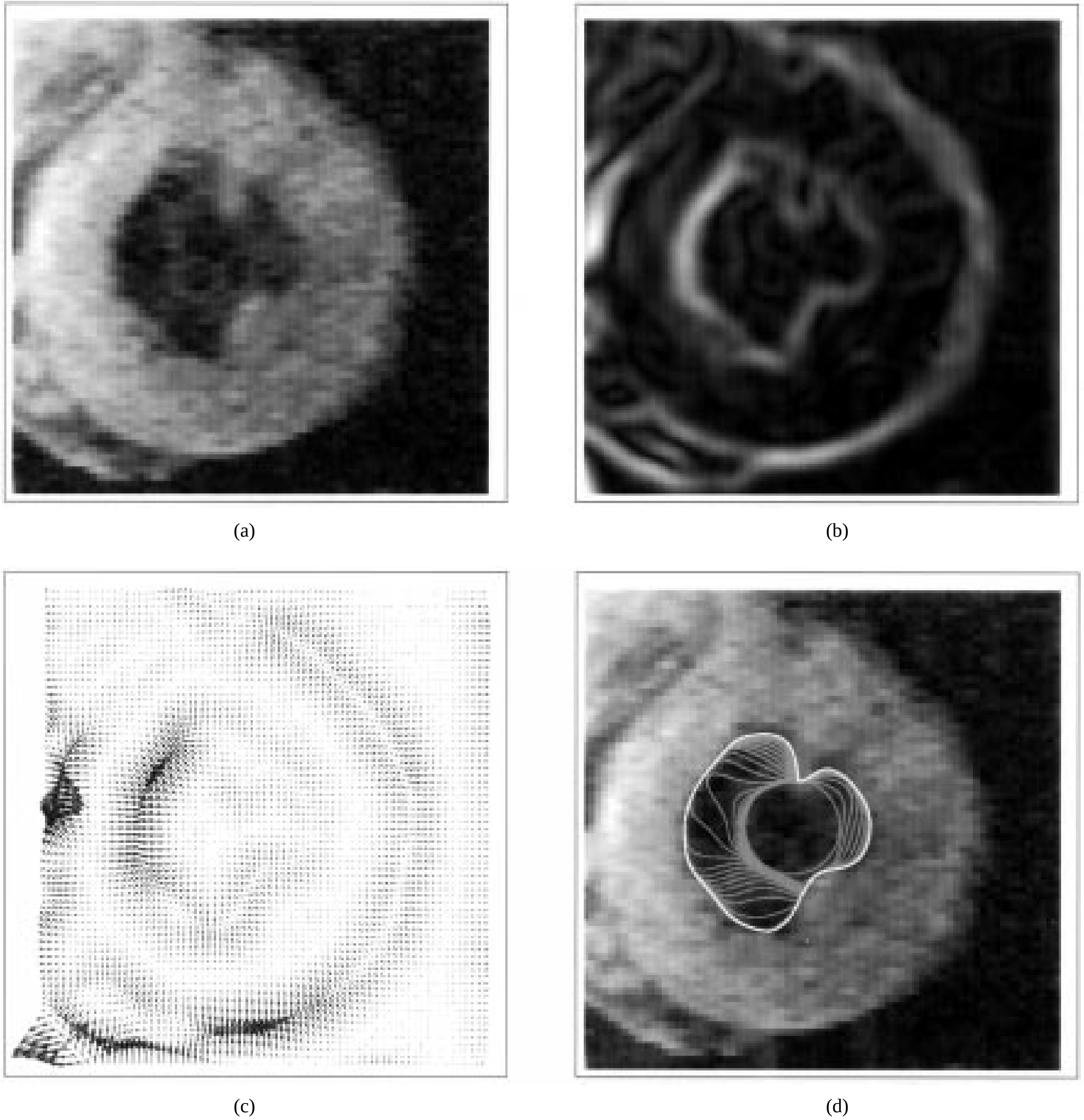


Fig. 8. (a) 160 × 160-pixel magnetic resonance image of the left ventricle of a human heart; (b) edge map $|\nabla(G_\sigma * I)|^2$ with $\sigma = 2.5$; (c) GVF field (shown subsampled by a factor of two); and (d) convergence of the GVF snake.

using $f(x, y) = f^{(2)}(x, y)$ with $\sigma = 1.5$ pixels, and Fig. 7(c) shows the computed GVF field. It is evident that the stronger edge-map gradients are retained while the weaker gradients are smoothed out, exactly as would be predicted by the GVF energy formulation of (12). Superposed on the original image, Fig. 7(d) shows a sequence of GVF snakes (plotted in a shade of gray) and the GVF snake result (plotted in white). The result shows an excellent convergence to the boundary, despite the initialization from far away, the image noise, and the boundary concavity.

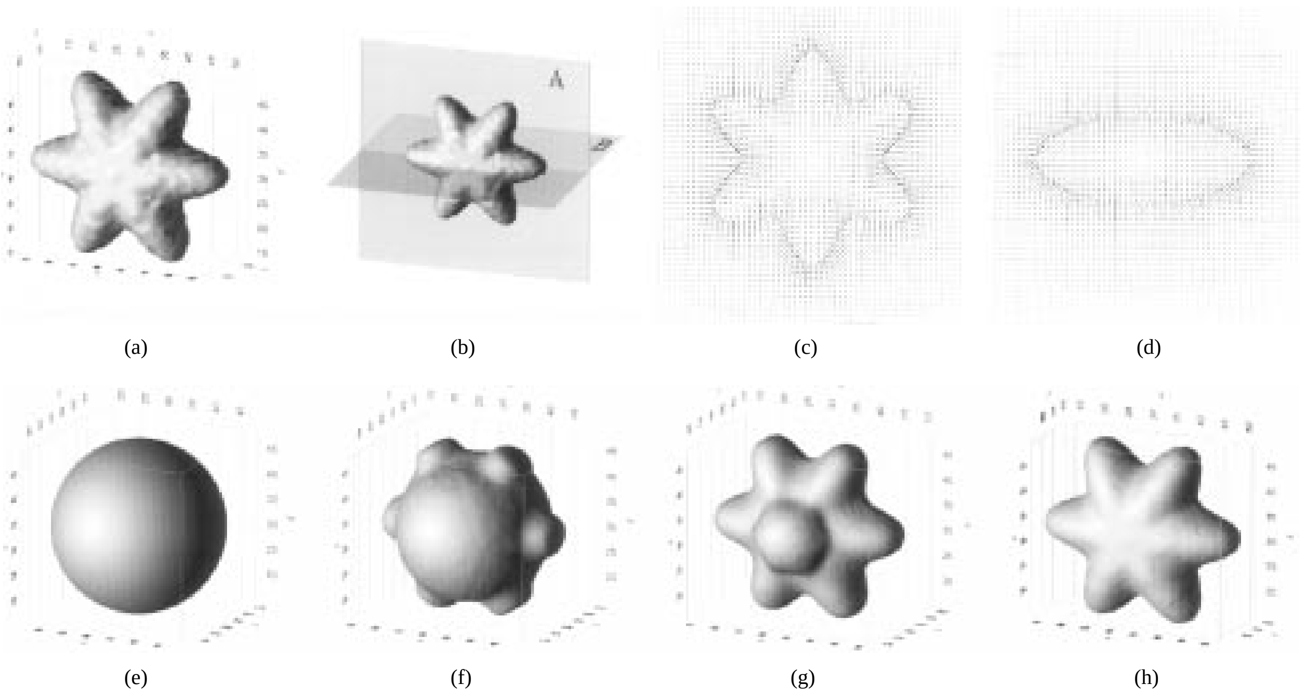
Another demonstration of GVF applied to gray-scale imagery is shown in Fig. 8. Fig. 8(a) shows a magnetic resonance image (short-axis section) of the left ventrical of a human heart, and Fig. 8(b) shows an edge-map computed using $f(x, y) = f^{(2)}(x, y)$ with $\sigma = 2.5$. Fig. 8(c) shows the computed GVF, and Fig. 8(d) shows a sequence of GVF snakes (plotted in a shade of gray) and the GVF snake result (plotted in white), both overlaid on the original image. Clearly, many details on the endocardial border are captured by the GVF snake result, including the papillary muscles (the bumps that protrude into the cavity).

B. Higher Dimensions

GVF can be easily generalized to higher dimensions. Let $f(\mathbf{x})$: $\mathbf{R}^n \to \mathbf{R}$ be an edge-map defined in $\mathbf{R}^n$. The GVF field in $\mathbf{R}^n$ is defined as the vector field $\mathbf{v}(\mathbf{x})$: $\mathbf{R}^n \to \mathbf{R}^n$ that minimizes the energy functional

$$\mathcal{E} = \int_{\mathbf{R}^n} \mu|\nabla\mathbf{v}|^2 + |\nabla f|^2\,|\mathbf{v} - \nabla f|^2\,d\mathbf{x} \tag{19}$$

where the gradient operator ∇ is applied to each component of $\mathbf{v}$ separately. Using the calculus of variations, we find that



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

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

Fig. 9. (a) Isosurface of a 3-D object defined on a 64^3 grid. (b) Positions of planes A and B on which the 3-D GVF vectors are depicted in (c) and (d), respectively. (e) Initial configuration of a deformable surface using GVF and its positions after (f) 10, (g) 40, and (h) 100 iterations.

the GVF field must satisfy the Euler equation

$$\mu\nabla^2\mathbf{v} - (\mathbf{v} - \nabla f)|\nabla f|^2 = \mathbf{0} \tag{20}$$

where ∇^2 is also applied to each component of the vector field $\mathbf{v}$ separately.

A solution to these Euler equations can be found by introducing a time variable t and finding the steady-state solution of the following linear parabolic partial differential equation

$$\mathbf{v}_t = \mu\nabla^2\mathbf{v} - (\mathbf{v} - \nabla f)|\nabla f|^2 \tag{21}$$

where $\mathbf{v}_t$ denotes the partial derivative of $\mathbf{v}$ with respect to t. Equation (21) comprises n decoupled scalar linear second order parabolic partial differential equations in each element of $\mathbf{v}$. Therefore, in principle, it can be solved in parallel. In analogous fashion to the 2-D case, finite differences can be used to approximate the required derivatives and each scalar equation can be solved iteratively.

A preliminary experiment using GVF in three dimensions was carried out using the object shown in Fig. 9(a), which was created on a 64^3 grid, and rendered using an isosurface algorithm. The 3-D GVF field was computed using a numerical approximation to (21) and $\mu = 0.15$. This GVF result on the two planes shown in Fig. 9(b), is shown projected onto these planes in Figs. 9(c) and (d). The same characteristics observed in 2-D GVF field are apparent here as well.

Next, a deformable surface (cf., [3]) using 3-D GVF was initialized as the sphere shown in Fig. 9(e), which is neither entirely inside nor entirely outside the object. Intermediate results after 10 and 40 iterations of the deformable surface algorithm are shown in Figs. 9(f) and 9(g). The final result after 100 iterations is shown in Fig. 9(h). The resulting surface is smoother than the isosurface rendering because of the internal forces in the deformable surface model.

VI. Summary and Conclusion

We have introduced a new external force model for active contours and deformable surfaces, which we called the gradient vector flow (GVF) field. The field is calculated as a diffusion of the gradient vectors of a gray-level or binary edge-map. We have shown that it allows for flexible initialization of the snake or deformable surface and encourages convergence to boundary concavities.

Further investigations into the nature and uses of GVF are warranted. In particular, a complete characterization of the capture range of the GVF field would help in snake initialization procedures. It would also help to more fully understand the GVF parameter μ, perhaps finding a way to choose it optimally for a particular image, and to understand the interplay between μ and the snake parameters α and β. Finally, the GVF framework might be useful in defining new connections between parametric and geometric snakes, and might form the basis for a new geometric snake.

Acknowledgment

The authors would like to thank D. Pham, S. Gupta, and Prof. J. Spruck for their discussions concerning this work, and the reviewers for providing several key suggestions for improving the paper.

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] T. McInerney and D. Terzopoulos, “A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis,” *Comput. Med. Imag. Graph.*, vol. 19, pp. 69–83, 1995.

[4] 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.

[5] R. Durikovic, K. Kaneda, and H. Yamashita, "Dynamic contour: A texture approach and contour operations," *Vis. Comput.*, vol. 11, pp. 277–289, 1995.

[6] D. Terzopoulos and R. Szeliski, "Tracking with Kalman snakes," in *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 3–20.

[7] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours," *Numer. Math.*, vol. 66, pp. 1–31, 1993.

[8] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–175, 1995.

[9] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *Proc. 5th Int. Conf. Computer Vision*, 1995, pp. 694–699.

[10] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image Understand.*, vol. 53, pp. 211–218, Mar. 1991.

[11] B. Leroy, I. Herlin, and L. D. Cohen, "Multi-resolution algorithms for active contour models," in *12th Int. Conf. Analysis and Optimization of Systems*, 1996, pp. 58–65.

[12] L. D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1131–1147, Nov. 1993.

[13] C. Davatzikos and J. L. Prince, "An active contour model for mapping the cortex," *IEEE Trans. Med. Imag.*, vol. 14, pp. 65–80, Mar. 1995.

[14] A. J. Abrantes and J. S. Marques, "A class of constrained clustering algorithms for object boundary extraction," *IEEE Trans. Image Processing*, vol. 5, pp. 1507–1521, Nov. 1996.

[15] C. Davatzikos and J. L. Prince, "Convexity analysis of active contour models," in *Proc. Conf. Information Science and Systems*, 1994, pp. 581–587.

[16] J. L. Prince and C. Xu, "A new external force model for snakes," in *Proc. 1996 Image and Multidimensional Signal Processing Workshop*, pp. 30–31.

[17] H. Tek and B. B. Kimia, "Image segmentation by reaction-diffusion bubbles," in *Proc. 5th Int. Conf. Computer Vision*, 1995, pp. 156–162.

[18] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 1997, pp. 66–71.

[19] P. M. Morse and H. Feshbach, *Methods of Theoretical Physics.* New York: McGraw-Hill, 1953.

[20] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[21] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.

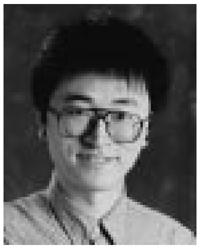
[22] S. N. Gupta and J. L. Prince, "Stochastic models for DIV-CURL optical flow methods," *IEEE Signal Processing Lett.*, vol. 3, pp. 32–35, 1996.

[23] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Interscience, 1953.

[24] A. H. Charles and T. A. Porsching, *Numerical Analysis of Partial Differential Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1990.

[25] W. F. Ames, *Numerical Methods for Partial Differential Equations*, 3rd ed. New York: Academic, 1992.

[26] S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," *IEEE Trans. Med. Imag.*, vol. 14, pp. 12–24, Mar. 1995.



Chenyang Xu (S'94) received the B.S. degree in computer science and engineering from the University of Science and Technology of China in 1993, and the M.S.E. degree in electrical and computer engineering from The Johns Hopkins University, Baltimore, MD, in 1995. He is currently a Ph.D. candidate in the Electrical and Computer Engineering Department, The Johns Hopkins University.

His research interests are in the areas of image processing, computer vision, medical imaging, and deformable models.

Jerry L. Prince (S'78–M'83–SM'96) received the B.S. degree from the University of Connecticut, Storrs, in 1979, and the S.M., E.E., and Ph.D. degrees in 1982, 1986, and 1988, respectively, from the Massachusetts Institute of Technology, Cambridge, all in electrical engineering and computer science.

After receiving the Ph.D. degree on the subject of geometric reconstruction in computed tomography, he joined the Technical Staff at Analytic Sciences Corporation, Reading, MA, where he contributed to the design of an automated vision system for synthetic aperture radar imaging. He joined the faculty at The Johns Hopkins University, Baltimore, MD, in 1989, and has been an Associate Professor in the Department of Electrical and Computer Engineering since 1994. He holds joint appointments in the Department of Radiology and the Department of Biomedical Engineering. His current research interests are in image processing and computer vision with primary application to medical imaging. Major projects include magnetic resonance imaging of cardiac motion, 3-D brain image analysis, and vector tomography.

Dr. Prince is a member of the Sigma Xi professional society and Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi honor societies. He is a former Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a recipient of the 1993 National Science Foundation Presidential Faculty Fellows Award and was recently selected as Maryland's 1997 Outstanding Young Engineer.