Gradient Vector Flow Deformable Models

Chenyang Xu[*] *and Jerry L. Prince*[*]

[*] Department of Electrical and Computer Engineering, The Johns Hopkins University,
3400 N. Charles St. Baltimore, MD 21218

KEY WORDS: medical imaging, image processing, deformable models, image segmentation

Send questions to:

Jerry L. Prince
105 Barton Hall
Johns Hopkins University
3400 North Charles Street
Baltimore, MD 21218
Tel: (410) 516-5192
Fax: (410) 516-5566
E-mail: prince@jhu.edu

Published in *Handbook of Medical Imaging*, Editor: Isaac Bankman, Academic Press, September 2000.

Abstract – Deformable models are used extensively in image processing, computer vision, and medical imaging applications, particularly to delineate object boundaries. Problems associated with initialization and poor convergence to boundary concavities, however, have limited their utility. This chapter presents an external force for deformable models, 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 deformable model external forces in that it cannot be written as the negative gradient of a potential function, and the corresponding deformable model is formulated directly from a dynamic force equation rather than a energy minimization formulation. Using several two-dimensional examples and two three-dimensional examples, we show that GVF has a large capture range and is able to move deformable models into boundary concavities.

1 Introduction

Deformable models are curves or surfaces defined within an image domain that can move under the influence of internal forces coming within the model itself and external forces computed from the image data. The internal and external forces are defined so that the model will conform to an object boundary or other desired features within an image. Deformable models are widely used in many applications, including edge detection [10, 5], shape modeling [18, 15], segmentation [12, 8], and motion tracking [19, 12].

There are two general types of deformable models in the literature today: *parametric deformable models* [10, 4, 18, 15] and *geometric deformable models* [2, 14, 3]. In this chapter, we focus on parametric deformable models, which synthesize parametric curves or surfaces within an image domain and allow them to move toward desired features, usually edges. Typically, the models are drawn toward the edges by *potential forces*, which are defined to be the negative gradient of potential functions. Additional forces, such as pressure forces [4], together with the potential forces comprise the *external forces*. There are also *internal forces* designed to hold the model together (elasticity forces) and to keep it from bending too much (bending forces).

There have been two key difficulties with parametric deformable models. First, the initial model 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 [4], and distance potentials [5]. The basic idea is to increase the *capture range* of the external force fields and to guide the model toward the desired boundary. The second problem is that deformable models have difficulties progressing into *boundary concavities* [7, 1]. There has been no satisfactory solution to this problem, although pressure forces [4], control points [7], domain-adaptivity [6], directional attractions [1], and the use of solenoidal fields [16] have been proposed. Most of the methods proposed to address these problems, however, solve only one problem while creating new difficulties. For example, multiresolution methods have addressed the issue of capture range, but specifying how the deformable model should move across different resolutions remains problematic. Another example is that of pressure forces, which can push an deformable model 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 chapter, we present a class of external force fields for deformable 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 solving a vector diffusion equation which diffuses the gradient vectors of a gray-level or binary edge map computed from the image. GVF was first introduced in [23] and a generalization to GVF was then proposed in [22]. In this chapter, we present the GVF in the context of its generalized framework. We call the deformable model that uses the GVF field as its external force a *GVF deformable model*. The GVF deformable model is distinguished from nearly all previous deformable model 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 dynamic force equation.

Particular advantages of the GVF deformable model over a traditional deformable model are its insensitivity to initialization and its ability to move into boundary concavities. As we show in this chapter, its initializations can be inside, outside, or across the object's boundary. Unlike deformable models that use pressure forces, a GVF deformable model does not need prior knowledge about whether to shrink or expand toward the boundary. The GVF deformable model 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 spatially varying diffusion process which does not blur the edges themselves, so multiresolution methods are not needed. The external force model that is closest in spirit to GVF is the distance potential forces of Cohen and Cohen [5]. 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 deformable model into boundary concavities. We believe that this is a property of all conservative forces which characterize nearly all deformable model external forces, and that exploring non-conservative external forces, such as GVF, is an important direction for future research in deformable models.

This chapter is organized as follows. We focus our most attention in 2-D and introduce the formulation for traditional 2-D parametric deformable models in Section 2. We next describe the 2-D GVF formulation in Section 3 and demonstrate its performance on both simulated and real images in Section 4. We then briefly present the formulation for 3-D GVF deformable models and their results on two examples in Section 5. Finally, in Section 6, we conclude this chapter and point out future research directions.

2 Background

2.1 2-D Parametric Deformable Models

A traditional 2-D parametric deformable model or deformable contour 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 deformable contour'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 potential 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 potential functions designed to lead a deformable contour toward step edges are [10]:

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

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

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 [4]:

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

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

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 deformable contour.

A deformable contour that minimizes E must satisfy the Euler equation [10]

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

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 deformable contour toward the desired image edges.

To find a solution to (6), the deformable contour 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. [10]). We note that most deformable contour implementations use either a parameter that multiplies $\mathbf{x}_t$ in order to control the temporal step-size, or a parameter that multiplies ∇E_{ext}, which permits separate control of the external force strength. In this chapter, 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.

2.2 Behavior of Traditional Deformable Contours

An example of the behavior of a traditional deformable contour is shown in Fig. 1. Fig. 1a 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 deformable contour ($\alpha = 0.6$, $\beta = 0.0$) initialized outside the object but within the capture range of the potential force field. The potential force field $\mathbf{F}_{\rm ext}^{(\rm p)} = -\nabla E_{\rm ext}^{(4)}$ where $\sigma = 1.0$ pixel is shown in Fig. 1b. We note that the final solution in Fig. 1a solves the Euler equations of the deformable contour formulation, but remains split across the concave region.

The reason for the poor convergence of this deformable contour is revealed in Fig. 1c, 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 deformable 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 deformable contour formulations, the problem of limited capture range, can be understood by examining Fig. 1b. 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 [5] proposed an external force model that significantly increases the capture range of a traditional deformable model. 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 2.1. Fig. 2 shows the performance of a deformable contour using distance potential forces. Fig. 2a shows both the U-shaped object (in gray) and a sequence of contours (in black) depicting the progression of the deformable contour from its initialization far from the object to its final configuration. The distance potential forces shown in Fig. 2b 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. 2a, this deformable contour 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. 2c. We see that, like traditional potential forces, these forces also point horizontally in opposite directions, which pulls the deformable contour 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 [5], 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.

3 GVF Deformable Contours

Our overall approach is to use the dynamic force equation (8) as a starting point for designing a deformable contour. We define below a novel external force field $\mathbf{v}(\mathbf{x})$ called *gradient vector flow* (GVF) field and replace the potential force $-\nabla E_{\text{ext}}$ in (8) with $\mathbf{v}(\mathbf{x})$, yielding

$$\mathbf{x}_t(s,t) = \alpha\mathbf{x}''(s,t) - \beta\mathbf{x}''''(s,t) + \mathbf{v}(\mathbf{x}) \tag{9}$$

We call the parametric curve solving the above dynamic equation a *GVF deformable contour*. It is solved numerically by discretization and iteration, in identical fashion to the traditional deformable contour [10].

Although the final configuration of a GVF deformable contour 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 $\mathbf{v}(\mathbf{x})$ can not, in general, be written as the negative gradient of a potential function. The loss of this optimality property, however, is well-compensated by the significantly improved performance of the GVF deformable contour.

3.1 Edge Map

We begin by defining an *edge map* $f(\mathbf{x})$ derived from the image $I(\mathbf{x})$ having the property that it is larger near the image edges.[1] We can use any gray-level or binary edge map defined in the image processing literature (cf. [9]); for example, we could use

$$f(\mathbf{x}) = -E_{\text{ext}}^{(i)}(\mathbf{x}) \tag{10}$$

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

[1] Other features can be sought by redefining $f(\mathbf{x})$ to be larger at the desired features.

edges. Second, these vectors generally have large magnitudes only in the immediate vicinity of the edges. Third, in homogeneous regions, where $I(\mathbf{x})$ is nearly constant, ∇f is nearly zero.

Now consider how these properties affect the behavior of a traditional deformable contour when the gradient of an edge map is used as an external force. Because of the first property, a deformable contour 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.

3.2 Gradient Vector Flow

We define the GVF field $\mathbf{v}(\mathbf{x})$ as the equilibrium solution to the following vector diffusion equation

$$\mathbf{u}_t = g(|\nabla f|)\nabla^2\mathbf{u} - h(|\nabla f|)(\mathbf{u} - \nabla f) \tag{11a}$$

$$\mathbf{u}(\mathbf{x}, 0) = \nabla f(\mathbf{x}) \tag{11b}$$

In Eq. (11a), the first term on the right is referred to as the *smoothing term* since this term alone will produce a smoothly varying vector field. The second term is referred as the *data term* since it encourages the vector field $\mathbf{u}$ to be close to ∇f computed from the data. The weighting functions $g(\cdot)$ and $h(\cdot)$ apply to the smoothing and data terms, respectively. Since these weighting functions are dependent on the gradient of the edge map which is spatially varying, the weights themselves are spatially varying, in general. Since we want the vector field $\mathbf{u}$ to be slowly-varying (or smooth) at locations far from the edges, but to conform to ∇f near the edges, $g(\cdot)$ and $h(\cdot)$ should be monotonically non-increasing and non-decreasing functions of $|\nabla f|$, respectively.

In [23], the following weighting functions were chosen:

$$g(|\nabla f|) \quad = \quad \mu \tag{12a}$$

$$h(|\nabla f|) \quad = \quad |\nabla f|^2 \tag{12b}$$

Since $g(\cdot)$ is constant here, smoothing occurs everywhere; however, $h(\cdot)$ grows larger near strong edges, and should dominate at the boundaries. Thus, GVF computed using such weighting functions should provide good edge localization. The effect of smoothing becomes apparent, however, when there are two edges in close proximity, such as when there is a long, thin indentation along the boundary. In this situation, GVF tends to smooth between opposite edges, losing the forces necessary to drive a deformable contour into this region.

To address this problem, in [22] we proposed weighting functions in which $g(\cdot)$ gets smaller as $h(\cdot)$ becomes larger. Then, in the proximity of large gradients, there will be very little smoothing, and the effective vector field will be nearly equal to the gradient of the edge map. There are many ways to specify such pairs of weighting functions. In [22], the following weighting functions were used:

$$g(|\nabla f|) = e^{-(\frac{|\nabla f|}{K})^2} \tag{13a}$$

$$h(|\nabla f|) = 1 - g(|\nabla f|) \tag{13b}$$

The GVF field computed using such weighting functions will conform to the edge map gradient at strong edges, but will vary smoothly away from the boundaries. The specification of K determines to some extent the degree of tradeoff between field smoothness and gradient conformity.

The vector diffusion equation (11) specifying GVF with various weighting functions, can be implemented using an explicit finite difference scheme described in [23], which is stable if the time step Δt and the spatial sample intervals Δx and Δy satisfy

$$\Delta t \leq \frac{\Delta x \Delta y}{4 g_{\max}}$$

where $g_{\max}$ is the maximum value of $g(\cdot)$ over the range of gradients encountered in the edge map image. While an implicit scheme for the numerical implementation of (11) would be unconditionally stable and therefore not need this condition, the explicit scheme is faster. Still faster methods — for example, the multigrid method — are possible.

In most examples, the use of either (12) or (13) produces very similar results. Here, we will demonstrate most of the properties of GVF using (12). When necessary to contrast their performance, we will refer to the GVF using (12) and (13) as GVF-I and GVF-II respectively.

4 Experiments

In this section, we first show several examples of GVF field computations on simple objects and demonstrate several key properties of GVF deformable contours. We then show the results of applying GVF deformable contours on both a noisy image and a real MR image. We used $\alpha = 0.6$ and $\beta = 0.0$ for all deformable contours and $\mu = 0.2$ for GVF unless stated separately. The deformable contours were dynamically reparameterized to maintain contour point separation to within 0.5–1.5 pixels (cf. [13]). All edge maps used in GVF computations were normalized to the range $[0, 1]$.

4.1 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. 3b, to the traditional potential force field of Fig. 1b, reveals several key differences. First, like the distance potential force field (Fig. 2b), the GVF field has a much larger capture range than traditional potential forces. A second observation, which can be seen in the closeup of Fig. 3c, 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. 1c and the distance potential forces of Fig. 2c. Finally, it can be seen from Fig. 3b 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. 3a shows the initialization, progression, and final configuration of a GVF deformable contour. The initialization is the same as that of Fig. 2a, and the deformable contour parameters are the same as those in Figs. 1 and 2. Clearly, the GVF deformable contour has a broad capture range and superior convergence properties. The final deformable contour configuration closely approximates the true boundary, arriving at a sub-pixel interpolation through bilinear interpolation of the GVF force field.

As discussed in Section 3.2, the GVF-I field tends to smooth between opposite edges when there is a long, thin indentation along the object boundary while the GVF-II field does not. Fig. 4 demonstrates this performance difference. Using an edge map obtained from the original image shown in Fig. 4a, both the GVF-I field ($\mu = 0.2$) and the GVF-II field ($K = 0.05$) were computed, as shown zoomed in Figs. 4b and 4c, respectively. We note that in this experiment both the GVF-I field and the GVF-II field were normalized with respect to their magnitudes and used as external forces. Next, a deformable contour ($\alpha = 0.25$, $\beta = 0$) was initialized at the position shown in Fig. 4d and allowed to converge within each of the external force fields. The GVF-I result, shown in Fig. 4e, stops well short of convergence to the long, thin, boundary indentation. On the other hand, the GVF-II result, shown in Fig. 4f, is able to converge completely to this same region. It should be noted that both GVF-I and GVF-II have wide capture ranges (which is evident because the initial deformable contour is fairly far away from the object), and they both preserve subjective contours (meaning that they cross the short boundary gaps).

4.2 Results on 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 as in the binary case.

Fig. 5a 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. 5b shows an edge-map computed using $f(x, y) = f^{(2)}(x, y)$ with $\sigma = 1.5$ pixels, and Fig. 5c shows the computed GVF field. It is evident that the stronger edge-map gradients are retained while the weaker gradients are smoothed out. Superposed on the original image, Fig. 5d shows a sequence of GVF deformable contours (plotted in a shade of gray) and the GVF deformable contour 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. 6. Fig. 6a shows a magnetic resonance image (short-axis section) of the left ventrical of a human heart, and Fig. 6b shows an edge map computed using $f(x, y) = f^{(2)}(x, y)$ with $\sigma = 2.5$. Fig. 6c shows the computed GVF, and Fig. 6d shows a sequence of GVF deformable contours (plotted in a shade of gray) and the GVF deformable contour result (plotted in white), both overlaid on the original image. Clearly, many details on the endocardial border are captured by the GVF deformable contour result, including the papillary muscles (the bumps that protrude into the cavity).

5 3-D GVF Deformable Models and Results

Both the GVF and the deformable contour formulations can be readily extended into 3-D. In fact, 3-D GVF has the identical formulation as the 2-D GVF described in Eq. (11) since it is written in a dimension-independent form. Examples of work on 3-D deformable models known also as deformable surfaces can be found in [5, 15].

Fig. 7 shows an experiment using a GVF deformable surface on a simulated 3-D image created on a 64^3 grid. The object to be reconstructed, rendered using an isosurface algorithm, is shown in Fig. 7a. The 3-D GVF field was computed using a numerical scheme similar to the one of 2-D with $\mu = 0.15$. This GVF result on the two planes shown in Fig. 7b, is shown projected onto these planes in Figs. 7c and d. The same characteristics observed in 2-D GVF field are apparent here as well. A deformable surface using 3-D GVF was initialized as the sphere shown in Fig. 7e, 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. 7f and g. The final result after 100 iterations is shown in Fig. 7h. The resulting surface is smoother than the isosurface rendering because of the internal forces in the deformable surface model.

Fig. 8 shows an example of using the GVF deformable surface to reconstruct a surface representation of the central layer of the human cerebral cortex from a 3-D MR brain image. Details of this work can be found in [21, 20].

6 Conclusions

We have introduced a novel external force model for deformable models, 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 deformable model 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 deformable model initialization procedures. It would also help to more fully understand the GVF-I parameter μ and the GVF-II parameter K, perhaps finding a way to choose them optimally for a particular image, and to understand the interplay between the GVF parameters and the deformable model parameters α and β. Also, the GVF framework might be useful in defining new connections between parametric and geometric deformable models, and might form the basis for a new geometric deformable model. Finally, making connections between GVF with other applications in image processing, computer vision, and medical imaging might provide some new insights or even new solutions to existing problems.

Acknowledgments

The authors would like to thank Dzung Pham, Sandeep Gupta, and Prof. Joel Spruck for their discussions concerning this work, and the support by National Science Foundation Presidential Faculty Fellow Award MIP93-50336 and by NIH Grant 1RO1NS37747-01.

References

[1] A. J. Abrantes and J. S. Marques. A class of constrained clustering algorithms for object boundary extraction. *IEEE Trans. on Image Processing*, 5(11):1507–1521, November 1996.

[2] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.

[3] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. Fifth Int. Conf. on Comp. Vis.*, pages 694–699, 1995.

[4] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[5] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Trans. on Pattern Anal. Machine Intell.*, 15(11):1131–1147, November 1993.

[6] C. Davatzikos and J. L. Prince. Convexity analysis of active contour models. In *Proc. Conf. on Info. Sci. and Sys.*, pages 581–587, 1994.

[7] C. Davatzikos and J. L. Prince. An active contour model for mapping the cortex. *IEEE Trans. on Medical Imaging*, 14(1):65–80, March 1995.

[8] R. Durikovic, K. Kaneda, and H. Yamashita. Dynamic contour: a texture approach and contour operations. *The Visual Computer*, 11:277–289, 1995.

[9] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, Engelwood Cliffs, NJ, 1989.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, 1(4):321–331, 1987.

[11] B. Leroy, I. Herlin, and L. D. Cohen. Multi-resolution algorithms for active contour models. In *12th International Conference on Analysis and Optimization of Systems*, pages 58–65, 1996.

[12] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. on Pattern Anal. Machine Intell.*, 15(6):617–634, 1993.

[13] S. Lobregt and M. A. Viergever. A discrete dynamic contour model. *IEEE Trans. on Medical Imaging*, 14(1):12–24, March 1995.

[14] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Pattern Anal. Machine Intell.*, 17(2):158–175, 1995.

[15] 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. *Computerized Medical Imaging and Graphics*, 19(1):69–83, 1995.

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

[17] H. Tek and B. B. Kimia. Image segmentation by reaction-diffusion bubbles. In *Proc. Fifth Int. Conf. on Comp. Vis.*, pages 156–162, 1995.

[18] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, 4:306–331, 1988.

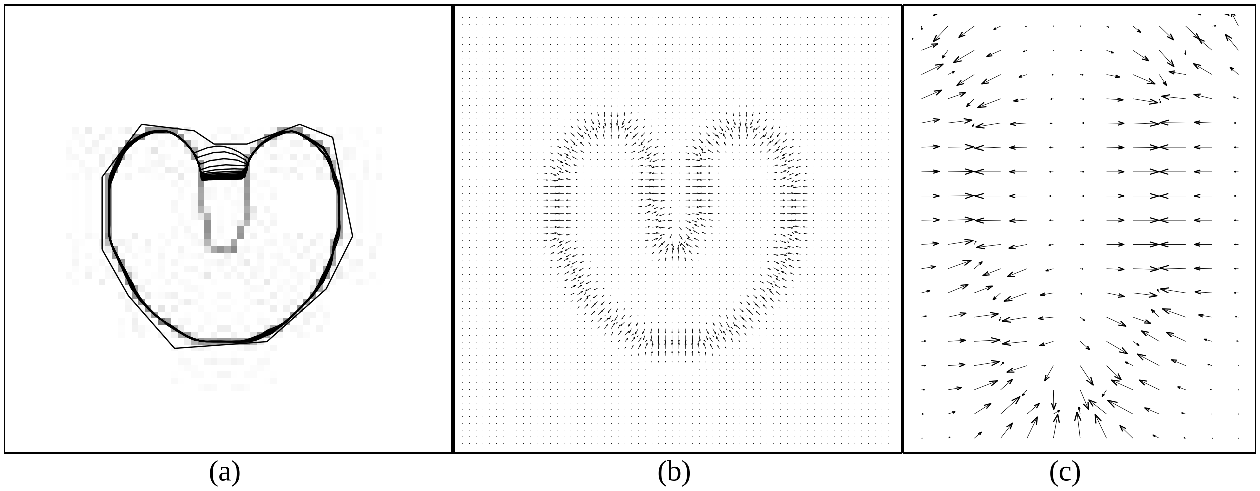


Figure 1: (a) The convergence of a deformable contour using (b) traditional potential forces, (c) shown close-up within the boundary concavity.

[19] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, Artificial Intelligence, pages 3–20. The MIT Press, Cambridge, Massachusetts, 1992.

[20] C. Xu. *Deformable Models with Application to Human Cerebral Cortex Reconstruction from Magnetic Resonance Images*. PhD thesis, The Johns Hopkins University, Baltimore, MD, USA, January 1999.

[21] C. Xu, D. L. Pham, J. L. Prince, M. E. Etemad, and D. N. Yu. Reconstruction of the human cortical surface from MR images. In *Proc. of the First International Conference on Medical Image Computing and Computer Assisted Interventions (MICCAI)*, pages 482–488, 1998.

[22] C. Xu and J. L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing, An International Journal*, 71(2):132–139, 1998.

[23] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. on Image Processing*, 7(3):359–369, March 1998.

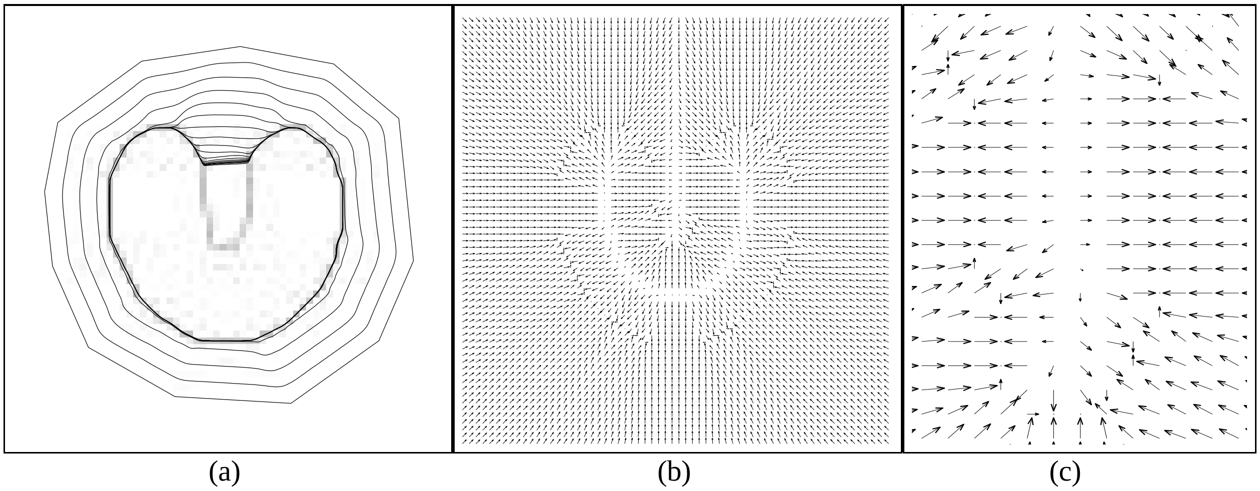


Figure 2: (a) The convergence of a deformable contour using (b) distance potential forces, (c) shown close-up within the boundary concavity.

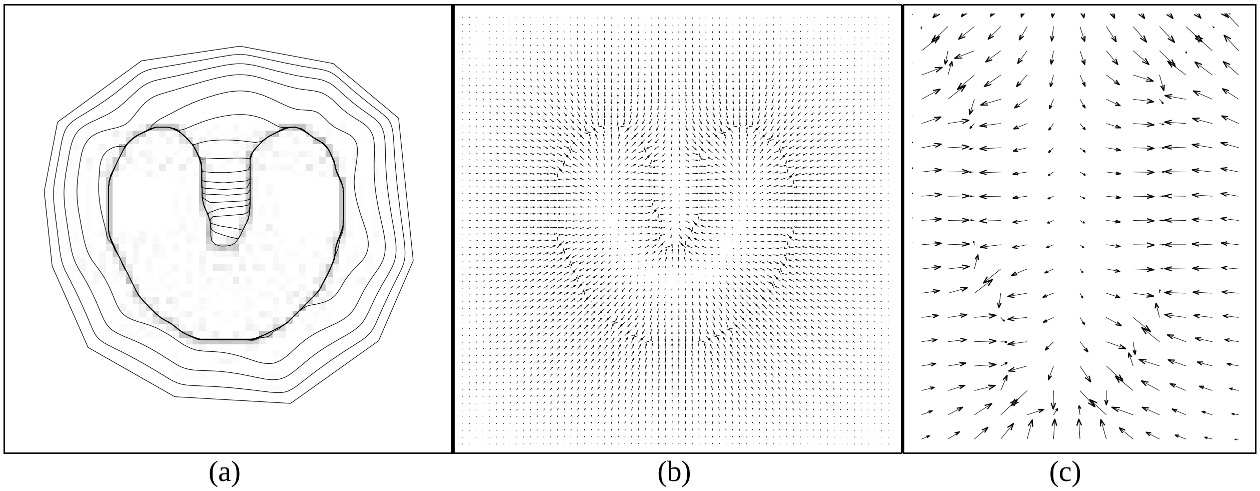
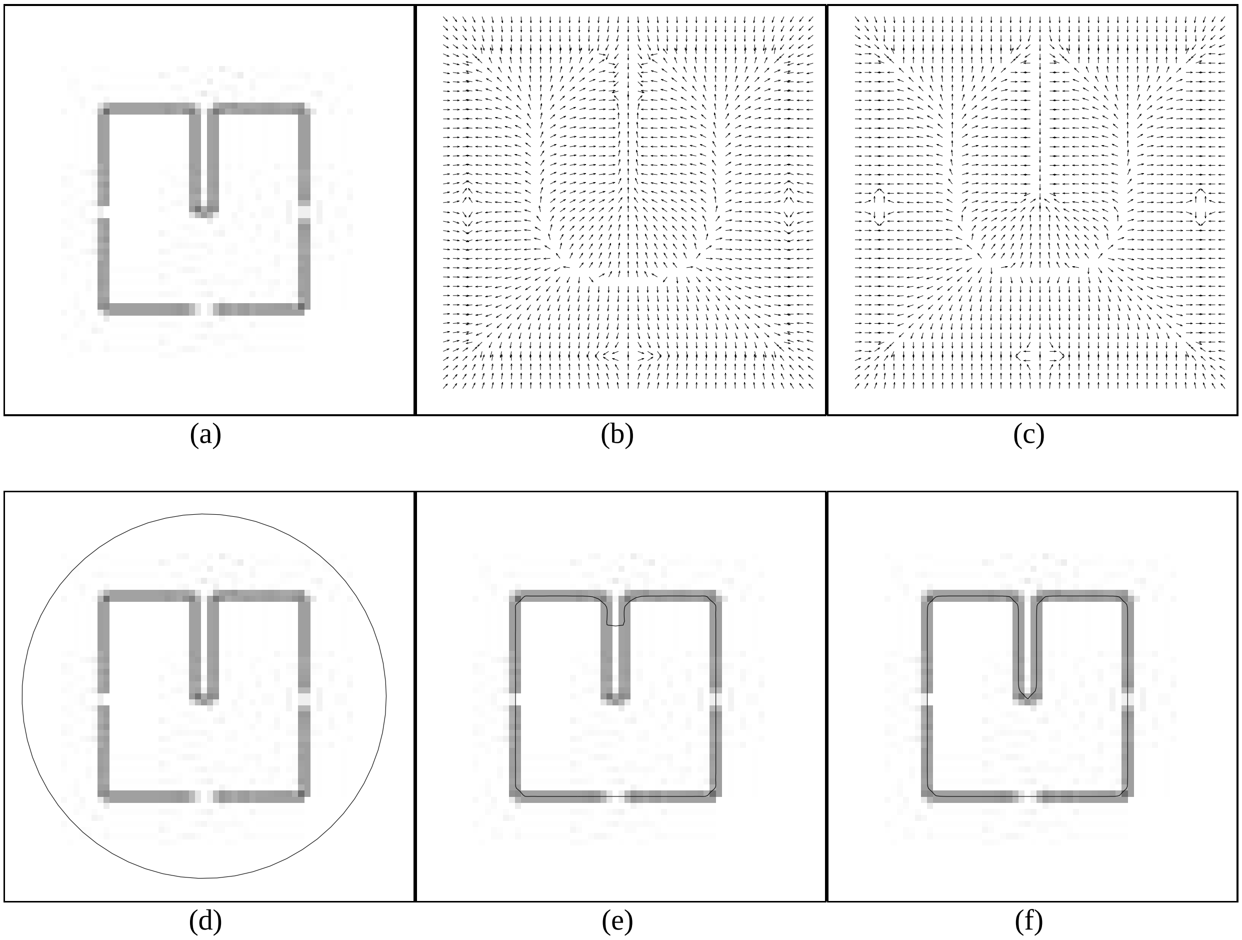


Figure 3: (a) The convergence of a deformable contour using (b) GVF external forces, (c) shown close-up within the boundary concavity.



(a) (b) (c)

(d) (e) (f)

Figure 4: (a) A square with a long, thin indentation and broken boundary; (b) GVF-I field (zoomed); (c) GVF-II field (zoomed); (d) initial contour position for both the GVF-I deformable contour and the GVF-II deformable contour; (e) final result of the GVF-I deformable contour; and (f) final result of the GVF-II deformable contour.

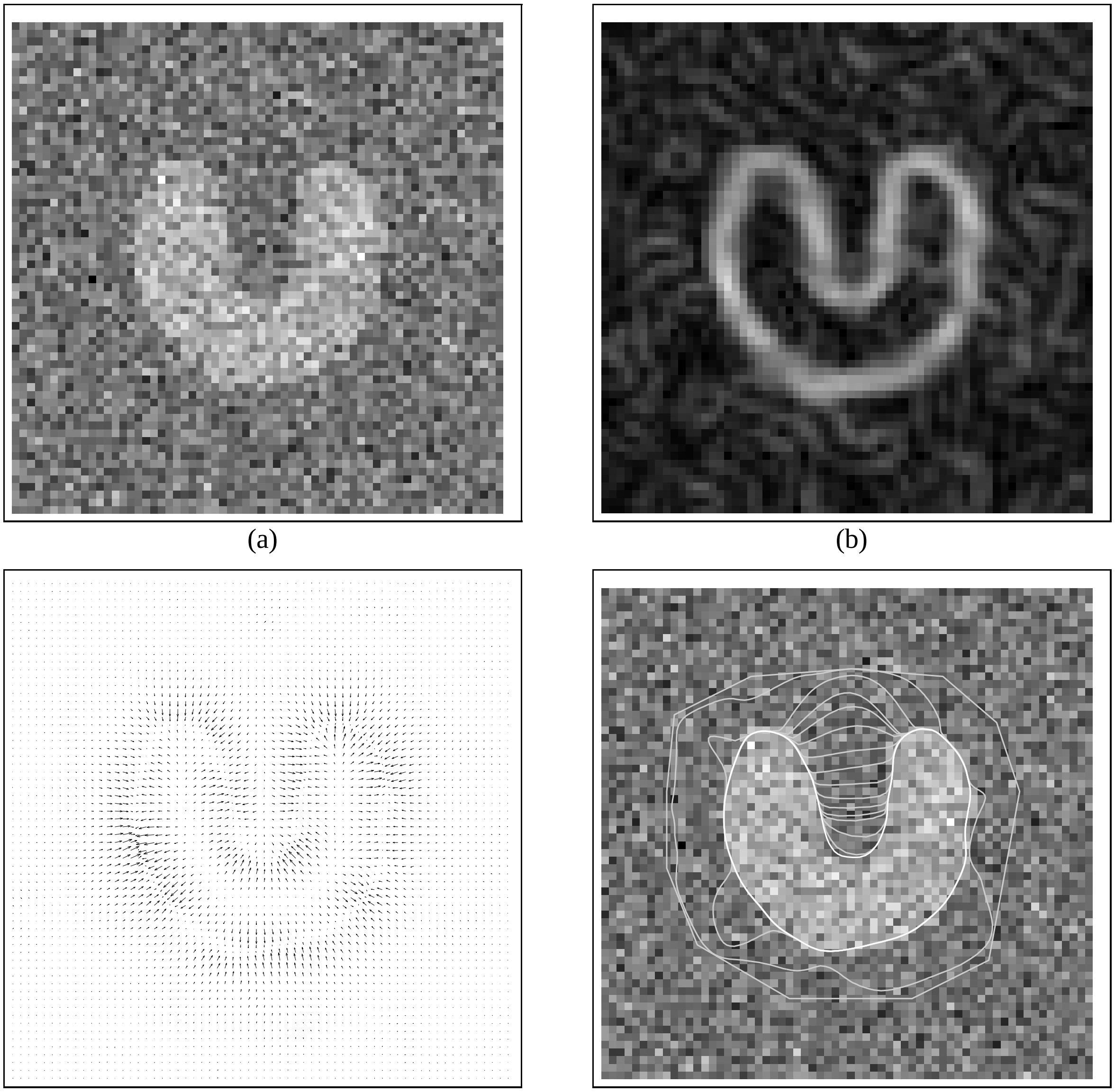
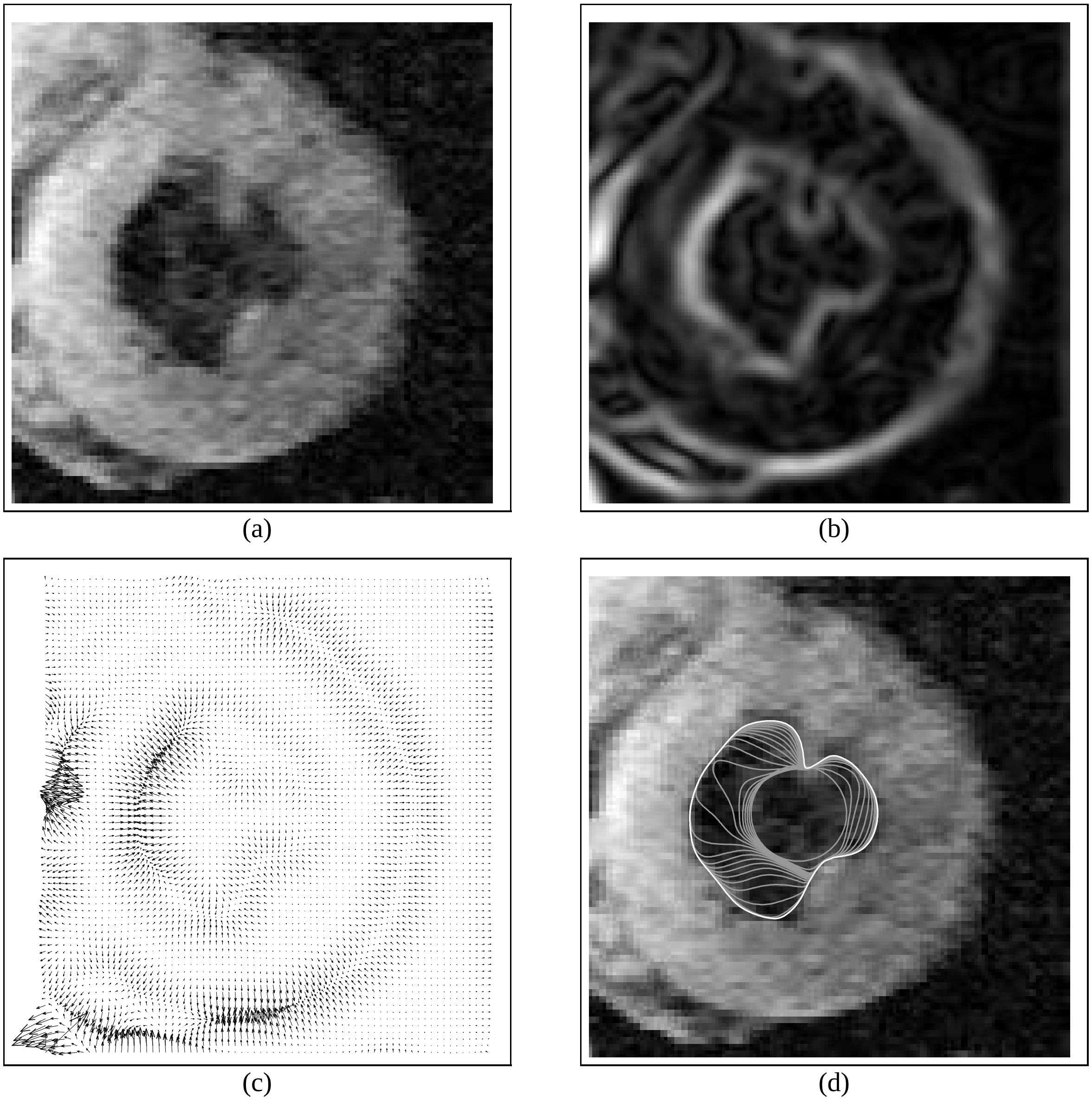
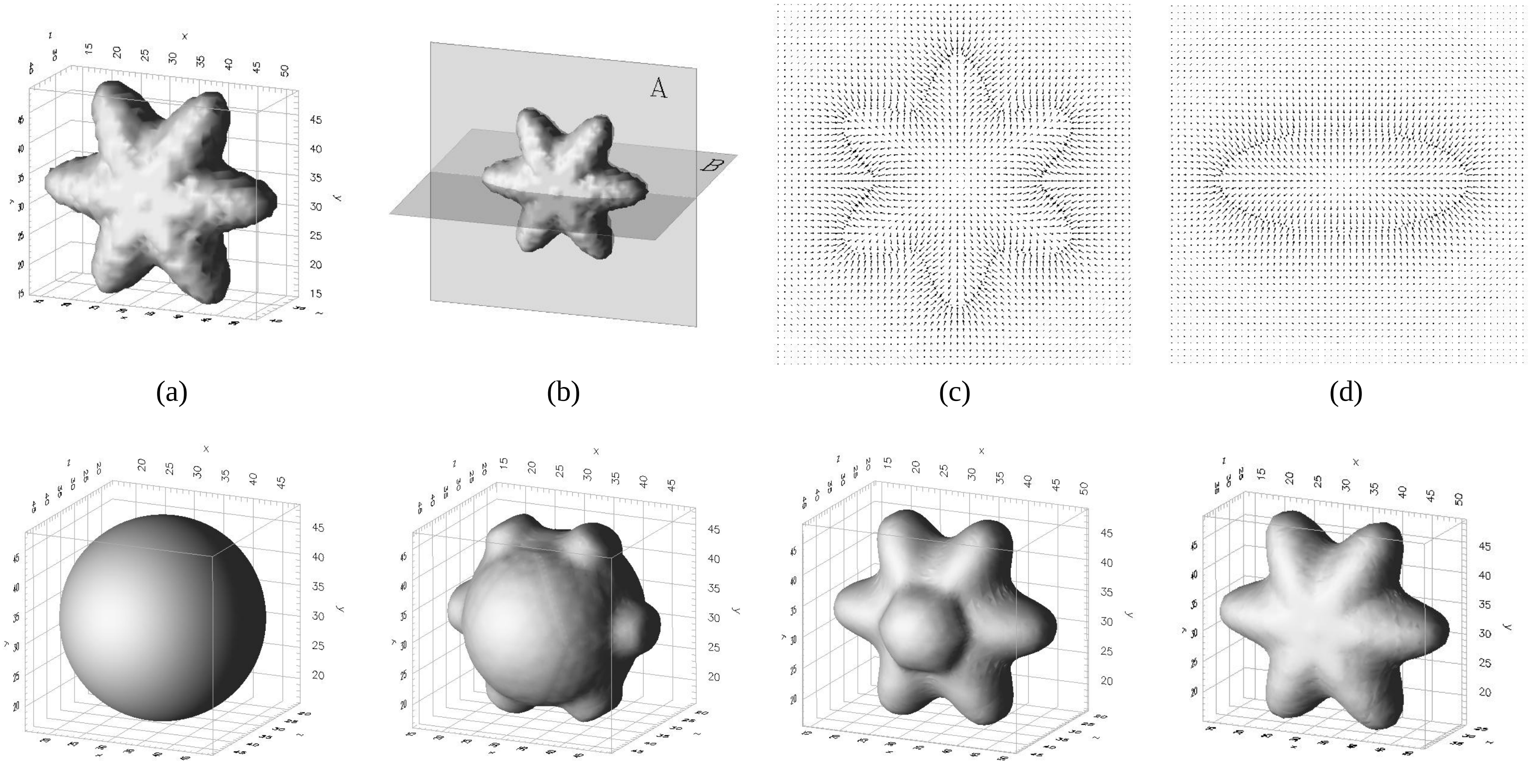


Figure 5: (a) A noisy 64×64-pixel image of a U-shaped object; (b) the edge map $|\nabla(G_\sigma * I)|^2$ with $\sigma = 1.5$; (c) the GVF external force field; and (d) convergence of the GVF deformable contour.



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

Figure 6: (a) A 160×160-pixel magnetic resonance image of the left ventrical of a human heart; (b) the edge map $|\nabla(G_\sigma * I)|$ with $\sigma = 2.5$; (c) the GVF field (shown subsampled by a factor of two); and (d) convergence of the GVF deformable contour.



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

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

Figure 7: (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) the initial configuration of a deformable surface using GVF and its positions after (f) 10, (g) 40, and (h) 100 iterations.

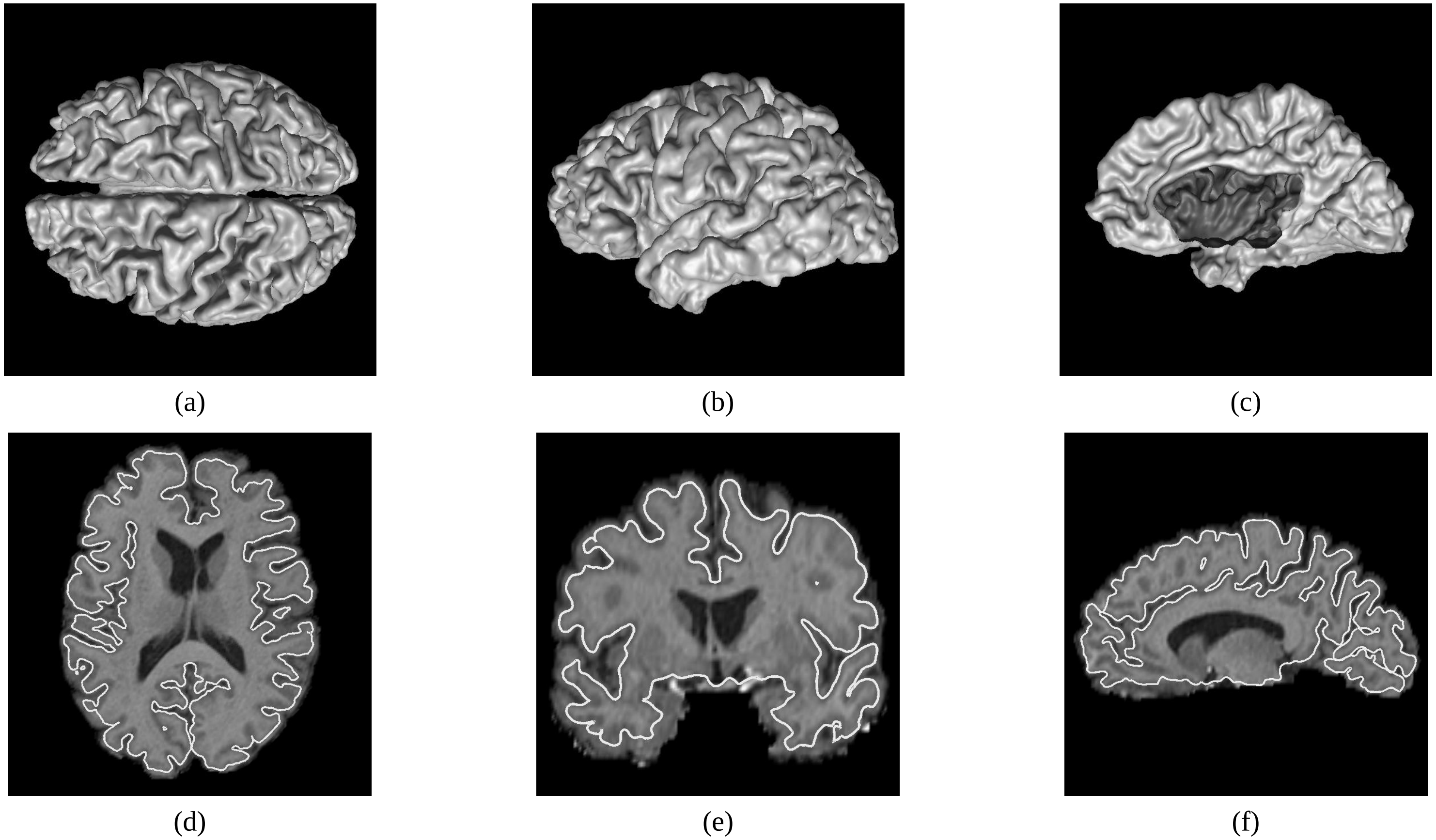


Figure 8: A surface rendering of reconstructed cortical surface from one subject displayed from multiple views: (a) top, (b) left, and (c) medial. Cross-sectional views of the same reconstructed cortical surface superimposed on the extracranial-tissues removed MR brain images: (d) axial, (e) coronal, and (f) sagittal.