IMAGE SEGMENTATION USING CURVE EVOLUTION

Baris Sumengen, B. S. Manjunath, and Charles Kenney
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
{sumengen, manj, kenney}@ece.ucsb.edu

ABSTRACT

A novel scheme for image segmentation is presented. The technique is based on the integration of ideas from geodesic active contours and a recently proposed edgeflow segmentation. Given an image a 2-D vector is constructed at each pixel location. This vector points in the direction of potential boundary pixels. The computation of the 2-D vector field is based on image intensity, color and texture gradients. Following this, an initial curve is instantiated and propagated to separate the image into foreground and background regions. The curve propagation is guided by the above mentioned vector field. The proposed approach thus utilizes an edge-based segmentation method and extends traditional PDE based curve evolution methods to texture image segmentation, and avoids the post-processing problems in edge linking and boundary detection.

1. INTRODUCTION

Image segmentation is a basic step in many image processing and computer vision tasks. Previous approaches to image segmentation include filtering-based methods to detect edges followed by edge linking [2], curve evolution and active contour models [1,4,11,14,15,16], region growing and merging [8], global optimization based on energy functions and Bayesian criteria [9], and graph partitioning and clustering [10]. Some of these methods seek to provide a unified framework that enables segmentation based on multiple heterogeneous attributes such as texture, color, and gray level intensity.

Curve evolution methods usually result in closed contours as opposed to disconnected edges resulting from filtering methods. However, their effectiveness in segmenting natural images that are rich in texture has not been clearly demonstrated. On the other hand some recent image segmentation methods have been successfully applied to a variety of images. Example for an edge-based method is the *edgeflow* technique [2], which uses a vector diffusion method to find edges. Our aim is to create a combined method that will result better segmentation result that can be applied to a vast variety of images.

The rest of the paper is organized as follows. We review active contour methods and edgeflow methods in section 2. In section 3, we present an edge-based hybrid approach to segmentation using edgeflow and geometric active contours. In section 4 we represent some experimental results and conclude with discussions and future work in section 5.

2. PREVIOUS WORK

Active contours and curve evolution methods usually define an initial contour C_0 and deform it towards the object boundary. The problem is usually formulated using partial differential equations (PDE). The previous research follows two different paths in terms of representation and implementation of active contours, namely parametric active contours (PACs) and geometric active contours (GACs). PACs use a parametric representation of the curves and GACs utilize level set methods [3,6]. Level set methods can easily handle topology changes of the evolving contour such as splitting and merging, and singularities on the curve such as sharp corners. Recently some connections between these two methods have been established [1, 7]. We will focus more on the previous work on the GACs in this paper. A summary and comparison of both GACs and PACs can be found in [7].

Curve evolution methods can be classified into several groups: edge-based [1,4,11], region-based [14,15] and hybrid [16] active contours.

Edge-based active contours aim to identify an object in an image by utilizing the local discontinuities of the image. The idea is to try to fit an initial closed contour to an edge function generated from the original image. The edges in this edge function are not connected, so they don't identify regions by themselves. An initial closed contour is slowly modified to fit on the nearby edges in an optimal way. Most of these methods require the curve to be initialized close to the real object boundary.

Let $C(\varphi):[0,1] \to \Re^2$ be a parameterization of a 2-D closed curve. A fairly general curve evolution can be written as:

$$\frac{\partial C}{\partial t} = (\alpha + \beta\kappa)\vec{N} + (\vec{S} \cdot \vec{N})\vec{N} \qquad (1)$$

where κ is the curvature of the curve, $\vec{N}$ is the normal vector to the curve, α, β are constants, and $\vec{S}$ is an underlying velocity field whose direction and strength depend on the time and position but not on the curve front itself. This equation will evolve the curve in the normal direction. The first term is a constant speed parameter that expands or shrinks the curve, second term uses the curvature to make sure that the curve stays smooth at all times and the third term guides the curve according to an independent velocity field.

In their independent and parallel works, Caselles et al. [11] and Malladi et al. [4] are among the first to use level set methods to extract objects from an image. They initialize a small curve inside one of the object regions and let the curve evolve until it reaches the object boundary. The evolution of the curve is controlled by the local gradient. This can be formulated by modifying (1) as:

$$\frac{\partial C}{\partial t} = g(F + \varepsilon\kappa)\vec{N} \qquad (2)$$

where F, ε are constants, and $g = 1/\left(1 + \left|\nabla \hat{I}\right|\right)$. $\hat{I}$ is the Gaussian smoothed image. If F is positive, the curve expands and if F is negative the curve shrinks. This is a pure geometric approach and the edge function, g, is the only connection to the image. The problem with this setup is that if the curve propagates beyond the desired boundary, there is no mechanism to attract the curve back to that object boundary.

Caselles et al. [1] introduced geodesic active contours, which is an improvement over the previous active contour methods. Starting with the *snakes* problem defined by Kass et al. [5], they reformulated the energy functional as a geodesic computation in a Riemannian space and found the following gradient descend equation:

$$\frac{\partial C}{\partial t} = g(F + \kappa)\vec{N} - (\nabla g \cdot \vec{N})\vec{N} \qquad (3)$$

Here ∇g defines a vector field on the pixels of the image. The corresponding vectors point normal towards the closest boundary or edge. The vectors are defined along a thin strip at both sides of the boundaries and their magnitude is zero or insignificant in other areas. The advantage of this method over the pure geometric approaches is that even if the curve propagates beyond the boundary, the ∇g term in (3) pulls the curve back towards the boundary. Even though an improvement over previous methods, this method is still prone to boundary leaking as shown in [17].

Edgeflow image segmentation [2] is a recently proposed method that is based on filtering and vector diffusion techniques. Its effectiveness has been demonstrated on a large class of images. It features multiscale capabilities and uses multiple image attributes such as intensity, texture or color. As a first step, a vector field is defined on the pixels of the image grid (Fig 1b). At each pixel, the vector's direction is oriented towards the closest image discontinuity at a predefined scale. The magnitude of the vectors depends on the strength and the distance of the discontinuity. After generating this vector field, a vector diffusion algorithm is applied to detect the edges. This step is followed by edge linking and region merging to achieve a partitioning of the image. Details can be found at [2].

3. COMBINING EDGEFLOW AND GAC

Much of the previous work on curve evolution has emphasized the geometrical nature of the segmentation problem while not paying attention to the diverse set of image attributes that need to be considered in segmenting an image. The recently proposed edgeflow method is quite effective on a large and diverse class of images, but requires post processing to detect closed contours. One of the contributions of our proposed method is to bring together the effectiveness of these two methods—the curve evolution and edgeflow techniques—in obtaining better segmentation results.

Most of the edge-based geometric active contours (GACs) make use of an edge function g and almost all of the active contours use an external force field $\vec{F}_{ext}$. The purpose of the edge function is to stop or slow down the evolving contour when it is close to an edge. So g is defined to be 0 on the edges and 1 on homogeneous areas. The external force $\vec{F}_{ext}$ is designed to attract the active contour towards the boundaries. At each pixel, the force vectors point towards the closest boundary on the image. Most of the research on parametric active contours (PACs) aims at designing external force fields to achieve better segmentation results. On the other hand in the formulation of GACs an edge function g is designed and the force field is usually generated as $\vec{F}_{ext} = \nabla g$ following the derivation of the geodesic active contours. So in the case of GACs, the external force field is tightly connected to the edge function g and the effort usually goes into the design of the edge function. Most commonly the edge function is defined as $g = 1/\left(1 + \left|\nabla \hat{I}\right|\right)$. It has been shown in [7] with comparison to its counterparts that custom designing $\vec{F}_{ext}$ can lead to better results and fix the shortcomings of geometric active contours such as boundary leaking. Only recently this external force field borrowed from PACs is integrated to GACs [12]. One of the shortcomings in the design of both edge functions and external forces is that they depend directly on the image gradients as the boundary locations even though it has been shown that the image gradient is very sensitive to the noise and is not very reliable.

We use the edgeflow vector field as our external force field in a geometric active contour formulation. Similar to other external force fields, edgeflow vectors also designed to point towards the closest boundaries. One of the advantages of edgeflow is that it doesn't depend directly on image gradients, it can be adjusted to different scales, and it is easily extendible to color and texture images.

Differing from the design of the GACs, we start with edgeflow vector field $\vec{S}$ as our force field, and generate an edge function V from it (Fig 1). On the other hand, unlike the formulation of PACs, we utilize an edge function in the curve evolution.

Having generated both the edge function and the external force field, our proposed curve evolution equation is

$$\frac{\partial C}{\partial t} = V\alpha\vec{N} + (\vec{S}\cdot\vec{N})\vec{N} + V\kappa\vec{N} \tag{4}$$

where α is a constant, V is an edge function 0 along the edges and 1 on the homogenous areas, $\vec{S}$ is the edgeflow vector field, κ is the curvature, and $\vec{N}$ is the normal to the curve.

Edgeflow vectors can also be calculated based on the image features such as pixel intensity, color, texture or combinations of them. Unlike other edge-based active contour methods, applying our segmentation method to texture or color images doesn't require any changes in the formulation of the curve evolution or in the implementation of it. This is because the flow vector calculation is separated from the curve evolution.

4. EXPERIMENTAL RESULTS

The implementation of our proposed method consists of two steps. In the first step, the edgeflow vectors and an edge function are generated. These outputs are used in the second step wherein a manually instantiated curve is propagated according to (4).

First, the edgeflow vectors are calculated using a predefined scale parameter. This vector field calculation is conducted using intensity, color, or texture features or a combination of them depending on the type of the image. For a detailed discussion of edgeflow computations we refer to [2]. After calculating the edgeflow vector field $\vec{S}$, the edge function V is derived as follows

$$V = \frac{1}{1+|\vec{S}|} \tag{5}$$

We use the well-known level set method formulation [3,6] to implement the curve evolution in (4). This requires defining a corresponding level set function U that embeds C as its zero level set and the time evolution of U. The level set equation corresponding to (4) is

$$\frac{\partial U}{\partial t} = V(\alpha+\kappa)|\nabla U| + \vec{S}\cdot\nabla U \tag{6}$$

Here U is a 3-D function where $U(x, y) = 0$ defines the evolving curve. U is generated from the initial curve using the signed distance function:

$$U(x,y,t=0) = \pm d \tag{7}$$

where d is the distance from (x,y) to C and the sign is chosen positive if (x,y) is outside the contour C and negative if inside the contour. At each iteration, the time step is normalized as

$$\Delta t \le 1/\text{abs}\max(V(\alpha+\kappa) + \vec{S}\cdot\nabla U/|\nabla U|) \tag{8}$$

to optimize for the speed of the convergence while keeping the stability by satisfying the Courant-Friedrichs-Levy (CFL) condition.

We have tested the segmentation method on different data sets. Segmentation result on a synthetic aperture radar image is shown in Fig. 2. Edgeflow vectors are calculated using texture features and an initial contour is shrunk to segment the object of interest.

Fig. 3 shows segmentation of a mammogram image. The objective is to extract the boundaries of the cyst in this image. The edgeflow vectors are calculated at three different scales ranging from a coarse scale to a fine scale, and the corresponding segmentation results are shown in Fig. 3(c-e).

Fig. 4 shows segmentation of a natural image using color and texture features. A small contour is initialized inside the tiger (Fig. 4b) and the corresponding segmentation result is shown in Fig. 4c. Another active contour is initialized on the background of the image. Corresponding curve evolutions and the segmentation result can be seen in Fig 4(e-g).

5. DISCUSSION

We have presented a semi-automated segmentation method using active contours framework with a variety of image features. Using texture and color features, active contours are successfully applied to a diverse set of images, such as synthetic aperture radar, medical and natural images. In our method, the vector field generation is separated from the curve evolution. This makes better designs of vector fields possible. We used edgeflow vector field in our implementation but any vector field with similar characteristics could be used such as [13]. We are currently investigating automated contour initialization and also the integration of image region properties with edge information.

Acknowledgments: This work was supported by an ONR grant #N00014-01-0391. We also like to thank Professor Chandrasekaran for many fruitful discussions.

6. REFERENCES

[1] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," International Journal of Computer Vision, Kluwer Academic Publishers, vol.22, pp. 61-79, Feb.-March 1997.

[2] Wei-Ying Ma, B.S. Manjunath, "EdgeFlow: a technique for boundary detection and image segmentation," IEEE Transactions on Image Processing, IEEE, vol.9, pp. 1375-88, Aug. 2000.

[3] S. Osher, J. A. Sethian, "Fronts propagating with Curvature dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," Journal of Computational Physics, 79, pp. 12-49, 1988.

[4] R. Malladi, J.A. Sethian, B.C. Vemuri, "Evolutionary fronts for topology-independent shape modeling and recovery" ECCV'94, Volume I, pp. 3-13, 1994.

[5] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: active contour models," International Journal of Computer Vision, vol.1, pp. 321-31, 1987.

[6] J. A. Sethian, "Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science", Cambridge University Press, 1999.

[7] C. Xu, A. Yezzi, L. Prince, "On the Relationship between Parametric and Geometric Active Contours", Technical Report JHU/ECE 99-14, Dec 1999.

[8] Hojjatoleslami, S.A., Kittler, J., "Region growing: a new approach", IEEE Transactions on Image Processing, vol.7, (no.7), July 1998. p.1079-84.

[9] Chang-Tsun Li, R. Wilson, "Image segmentation based on a multiresolution Bayesian framework", Proceedings 1998 International Conference on Image Processing, p.761-5 vol.3

[10] Jianbo Shi, J. Malik, "Normalized cuts and image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.22, (no.8), Aug. 2000. p.888-905

[11] V. Caselles, F. Catte, T. Coll, F. Dibos, "A geometric model for active contours in image processing", Numerische Mathematik, vol.66, (no.1), 1993. p.1-31

[12] N. Paragios, O. Mellina-Gottardo, V. Ramesh, "Gradient Vector Flow Fast Geodesic Active Contours", ICCV 2001.

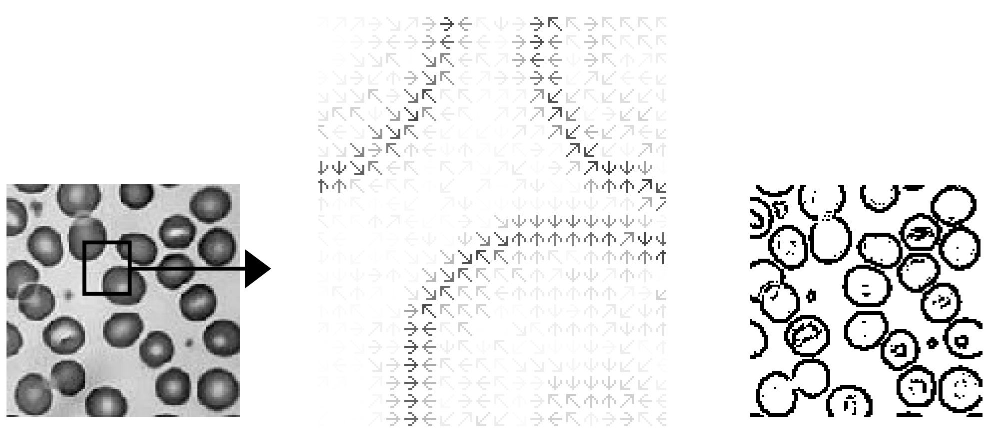
[13] M. Tabb, N. Ahuja, "Multiscale image segmentation by integrated edge and region detection", IEEE Transactions on Image Processing, vol.6, (no.5), May 1997. p.642-55.

[14] A. Tsai, A. Yezzi, A.S. Willsky, "A curve evolution approach to smoothing and segmentation using the Mumford-Shah functional", Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000, p.119-24.

[15] T.F. Chan, L.A. Vese, "Active contours without edges", IEEE Transactions on Image Processing, vol.10, (no.2), Feb. 2001. p.266-77.

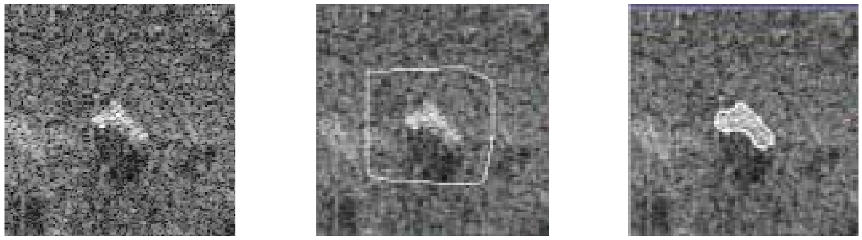
[16] N. Paragios, R. Deriche, "Geodesic active regions for supervised texture segmentation", Proceedings of the Seventh IEEE International Conference on Computer Vision, (vol.2), p.926-32.

[17] K. Siddiqi, Y.B. Lauziere, A. Tannenbaum, S.W. Zucker, "Area and length minimizing flows for shape segmentation", IEEE Transactions on Image Processing, vol.7, (no.3), March 1998, p.433-43.



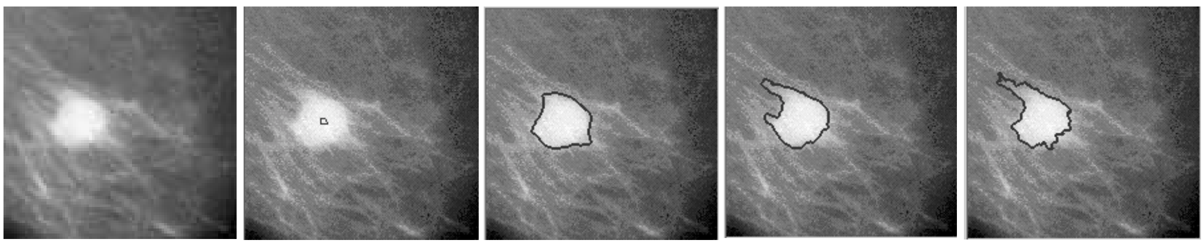
(a) (b) (c)

Fig. 1. (a) Image of blood cells. (b) Edgeflow vector field corresponding to the rectangle on the image. (c) Edge function of the image generated from edgeflow vectors.



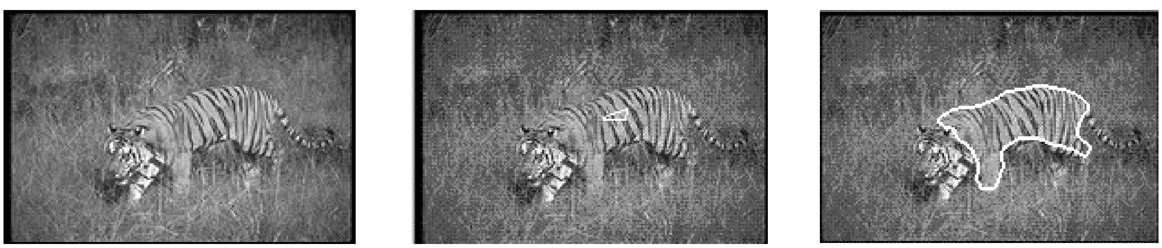
(a) (b) (c)

Fig. 2. (a) SAR image. (b) Initial contour. (c) Final boundary.

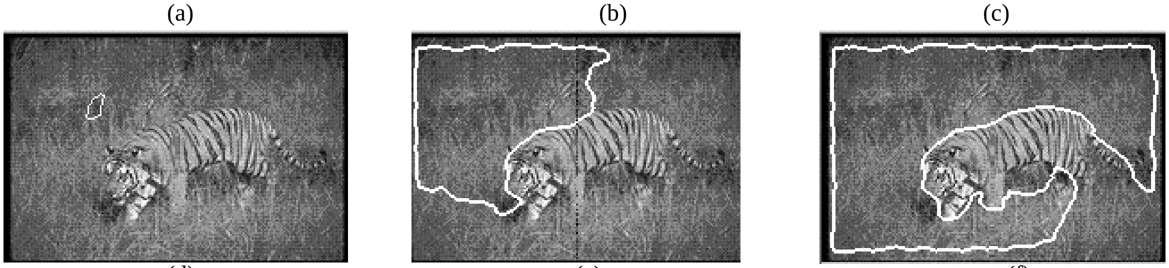


(a) (b) (c) (d) (e)

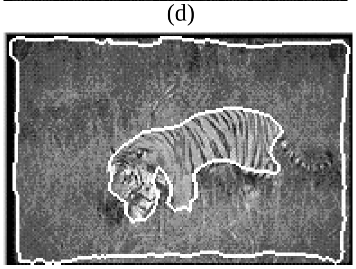
Fig. 3. (a) A mammogram image. (b) A contour is initialized inside the cyst. (c-e) Segmentation results corresponding to three different scales ranging from coarse to fine.



(a) (b) (c)



(d) (e) (f)



(g)

Fig. 4. Segmentation results of a natural image (a) using color and texture features. (b) A contour is initialized inside the tiger. (c) Active contour expands and finds the object boundary. (d) A contour is initialized on the background. (e-f) Evolution of the contour while it expands. (g) Final segmentation result.