Nonparametric Snakes

Umut Ozertem, *Student Member, IEEE*, and Deniz Erdogmus, *Member, IEEE*

***Abstract*—Active contours, or so-called *snakes*, require some parameters to determine the form of the external force or to adjust the tradeoff between the internal forces and the external forces acting on the active contour. However, the *optimal* values of these parameters cannot be easily identified in a general sense. The usual way to find these required parameters is to run the algorithm several times for a different set of parameters, until a satisfactory performance is obtained. Our nonparametric formulation translates the problem of seeking these unknown parameters into the problem of seeking a good edge probability density estimate. Density estimation is a well-researched field, and our nonparametric formulation allows using well-known concepts of density estimation to get rid of the exhaustive parameter search. Indeed, with the use of kernel density estimation these parameters can be defined locally, whereas, in the original snake approach, all the shape parameters are defined globally. We tested the proposed method on synthetic and real images and obtained comparatively better results.**

***Index Terms*—Active contours, image segmentation, kernel density estimation, nonparametric methods, snakes.**

I. Introduction

IMAGE segmentation is one of the fundamental problems in image processing. The most common image segmentation applications include feature extraction from images, filtering of noisy images, object recognition, and object based video or image coding. The definition of the image segmentation problem is as follows: Partition the image into distinct regions in a way that these regions are homogenous, and none of the unions of these distinct regions form a homogenous structure. The homogeneity can be defined in any feature space; hence, it does not strictly require homogeneity in the color or intensity space.

Existing image segmentation methods can be categorized into few main groups. The earliest methods are the *edge-based approaches* [1]. Edge-based approaches first detect the edges of the image, which are subsequently connected and combined to build object contours. However, these methods are efficient only if the pixel intensity itself is a suitable feature for segmentation and the main disadvantage of the edge-based methods is the fact that they are applicable only when the edges can be computed, which is not always possible. There are *region-based approaches*, like region growing, that have the advantage of low computational cost [2]. However, region-based methods are parametric, and the results of these methods suffer from being very sensitive to the parameter values. Another category is *split and merge approach* that partitions the image into primitive regions and merges them to provide the final results. A similarity measure is defined to compare similarities between the pairs of neighboring regions to merge them until a stopping criteria is ensured [3]. The problem of setting good thresholds and selecting an effective stopping criterion is mostly overcome by utilization of clustering methods for image segmentation problems. *Clustering-based methods* bring the strength of generalization properties of unsupervised learning problems into the image segmentation area [4], [5]. These approaches are generally nonparametric or have a few parameters, whose efficient values can usually be estimated from the data.

Another approach that has evolved from edge based approaches is the active contours, so called *snakes* [6]–[8]. Snakes are based on the utilization of the shape priors with the gradient of the edge map of the image. Active contours move with the effect of internal and external forces. Internal forces, such as elasticity forces or bending forces, depend on the contour itself, and the external forces are independent of the shape of the contour and are only evaluated from the image intensities. These forces are determined in a way that they will make the snake move towards the object boundaries. The external energy function is defined such that it takes its optimal (generally minimum, depending on the definition of the optimization problem) values at the object boundaries, while the internal energy function preserves the shape of the snake. Generally, in parametric and geometric active contours, edge maps or some derivatives of edge maps are used to define the external energy function. A different track in active contours is the *region-based active* contours [9], which are based on the idea of making the contour move towards a boundary such that both the interior and the exterior of the contour form homogenous regions.

There are several practical difficulties in using snakes, with the original definition, most of which are later addressed by the *gradient vector flow* (GVF) snake [8]. The most important problem is that the initialization of the active contour has to be in a narrow neighborhood of the object boundary. The range of initializations that leads to the desired segmentation is known as the *capture range* of the corresponding external force field. To overcome the low capture range problem in real life scenarios, snakes are usually initialized using the output of another segmentation algorithm, in which case, snakes can be regarded as a fine tuning step for another segmentation algorithm. To increase the capture range, several methods have been proposed [10]–[12]. A multiresolution based method is proposed by Leroy and coauthors [11], which addresses the capture range issue. However, determining the way the snake should move

Manuscript received September 2, 2006; revised May 19, 2007. This work was supported in part by the National Science Foundation under Grants ECS-0524835 and ECS-0622239. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mario A. T. (G. E.) Figueiredo.

The authors are with OGI, Beaverton, OR 97006 USA (e-mail: ozertemu@csee.ogi.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2007.902335

through varying resolution levels remains unsolved. Another approach to solve the same problem is *distance potentials method* proposed by Cohen and Cohen [12], which introduces an external force model to guide the contour towards the object boundary. This approach significantly increases the capture range and overcomes the initialization difficulty in most of the cases.

Another well-known problem is that the snakes have difficulties in progressing into concavities along the boundary. Directional attractions [14], control points [13], and pressure forces [10] are among the methods proposed to solve this problem. However, the most satisfactory results are obtained with the GVF formulation by Xu and Prince [8], which also solves the capture range issue very effectively. *GVF snakes* provide a principled and effective way of defining the external field, providing an insensitivity to initialization and an ability to progress into boundary concavities. However, the method of defining the internal energy function with the optimal selection of its parameters remains to be determined. As with most parametric methods, the usual way of seeking the desired result is to run the algorithm several times for a set of different parameter values until a satisfactory performance is obtained.

We approached the problem of defining the energy function in a nonparametric way. Our approach translates the problem of seeking efficient values for the parameters into the problem of seeking for an efficient density estimate. Density estimation is a well-researched field, and exploiting the connection that we use, different density estimation methods can be applied. Here, we mainly focused on kernel density estimation (KDE) [15] and derived an algorithm that exploits the underlying kernel density estimate of the data.

In the next section, we start with how we employ KDE to define the nonparametric snake, also emphasizing the reasons behind using KDE in the density estimation step. We will continue with a fast and practical algorithm and present our results providing comparisons in several synthetic and real data examples.

II. Nonparametric Snakes

In this section, we develop the nonparametric snake. In this particular realization of the concept, kernel density estimation, Euclidean inner product, and fixed-point iterations are employed. A discussion of how to address typical problems encountered by active contours in the proposed nonparametric framework will be presented.

In general, determining a suitable parametric family is problematic. Data probability density functions may take complex forms, hence, nonparametric density estimation methods are preferred for their versatility. KDE offers continuous and differentiable density estimates provided that continuous and differentiable kernel functions are employed. Further, exploiting variable kernel bandwidth parametrizations provide further benefits similar to nearest neighbor techniques while maintaining smoothness.

Determining a suitable kernel function is the most significant step in KDE, and there is a wide literature about how to select the kernel function [15]–[17]. Usually, isotropic or anisotropic Gaussians are selected as the kernel function, which will suffice for our purposes. The most typical choices for the kernel bandwidth are *i)* Silverman's rule of thumb, *ii)* leave-one-out cross validation maximum likelihood estimator, *iii)* median of the K nearest neighbor distance with an isotropic Gaussian, and *iv)* covariance of the K nearest neighbor distance with a anisotropic Gaussian. The first two examples are utilized for identifying optimal fixed-bandwidth kernels under minimum-squared-error and Bayesian reasoning, whereas the latter two represent variable-bandwidth kernel selection heuristics commonly utilized in similarity based learning techniques.

Selecting the kernel bandwidth according to an objective function is a well-researched field. Therefore, throughout the paper, we will use the term *suitable kernel function*, leaving the specific selection to the user in the broader context. Specifics of kernel selection in the presented experiments will be provided. There exist variable-weight variants of KDE, as well, though they are not as popular as variable bandwidth. The most general variable-weight variable-anisotropic-bandwidth KDE will be the generic tool that will be exploited in the proposed framework.

A. Nonparametric Edge Density Estimates

Consider an image $I(x,y)$. For each pixel the vector $\mathbf{s} = [x,y]^T$ denotes its location. Let $E(\mathbf{s})$ denote the edge map of this image as a function of space. The edge maps can be obtained using any suitable edge detector of choice and the edge values can be binary or continuous.

1) Estimating the Edge Density From a Binary Edge Map: In order to illustrate the edge distribution estimates obtained using KDE, consider the more intuitive case of a binary edge map $E(\mathbf{s})$ such that

$$E(\mathbf{s}) = \begin{cases} E(\mathbf{s}_i) = 1: & \mathbf{s}_i \text{ is an edge pixel} \\ E(\mathbf{s}_i) = 0: & \text{otherwise.} \end{cases} \tag{1}$$

A KDE of the edge map can be constructed using variable-anisotropic-bandwidth kernels as follows:

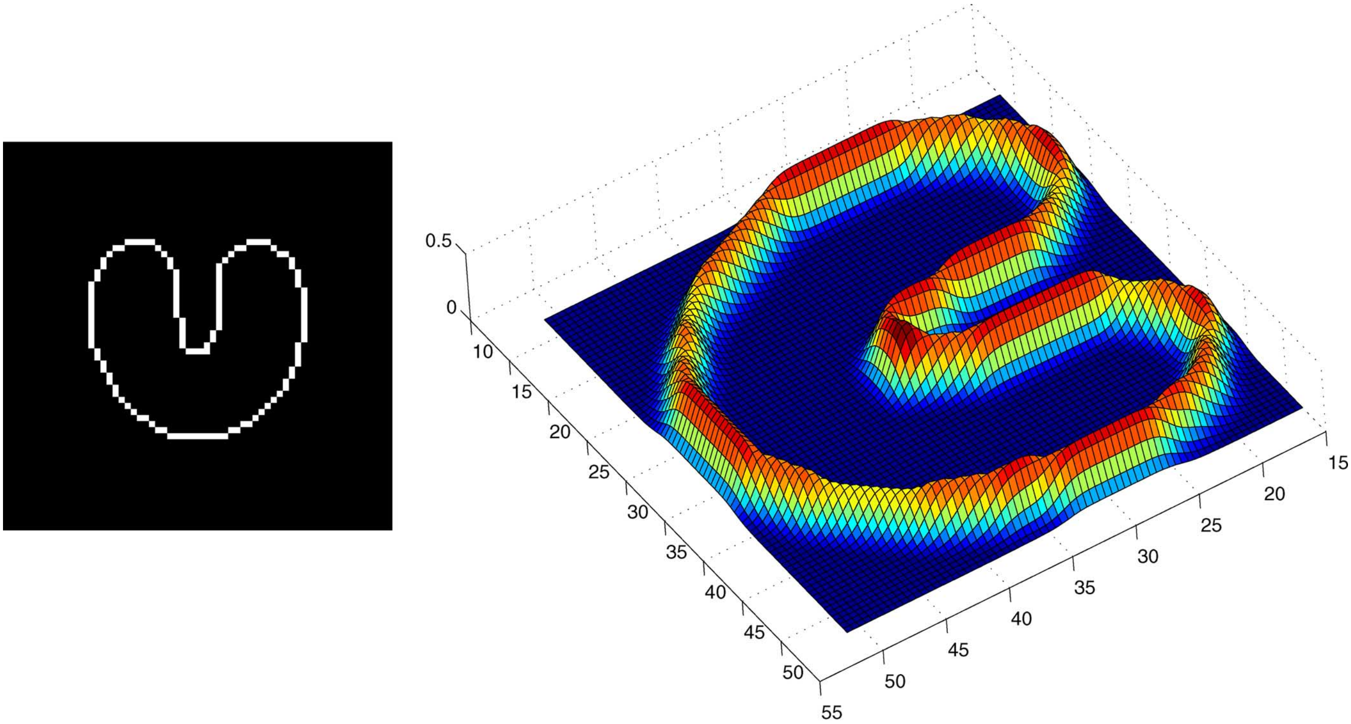
$$p_{\text{edge}}(\mathbf{s}) = \frac{1}{N_{\text{edge}}} \sum_{i=1}^{N} E(\mathbf{s}_i) K_{\Sigma_i}(\mathbf{s} - \mathbf{s}_i^{\text{edge}}) \tag{2}$$

where N is the number of pixels and

$$N_{\text{edge}} = \sum_{i=1}^{N} E(\mathbf{s}_i). \tag{3}$$

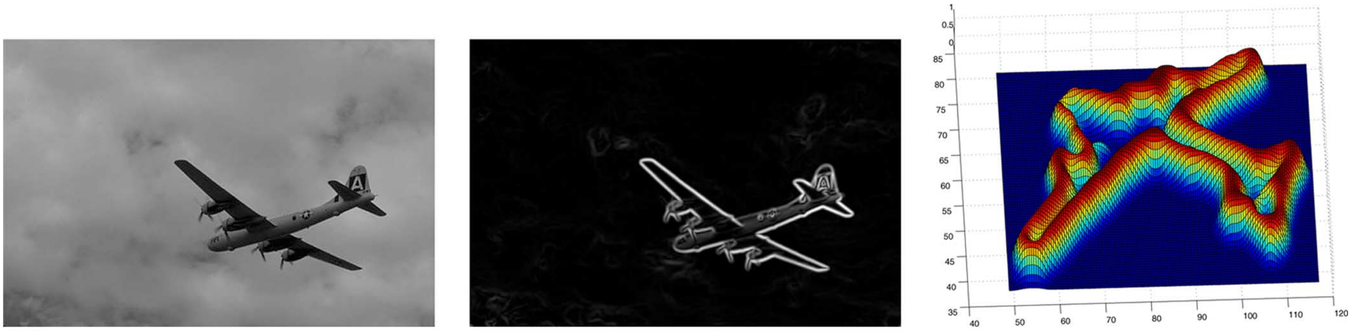
An illustration of this density estimate using fixed-isotropic-bandwidth Gaussian kernels is shown for the simple *U-shaped image*, in Fig. 1. This image has also been used by Xu and Prince to underline the GVF snake's ability to progress towards the boundary concavities [8]. The same will be illustrated for the nonparametric snake on this image later.

2) Estimating the Edge Density From a Continuous Edge Map: Suppose we have a continuous edge map $E_{\text{cont}}(\mathbf{s}) \in [0,\infty)$. The most common choices for a continuous edge map include: *i)* a binary edge map convolved with a Gaussian, *ii)* the magnitude-square of the gradient field, and *iii)* the magnitude-square of the gradient field convolved with a Gaussian.



(a) (b)

Fig. 1. Example of the edge probability density estimate for the binary case. (a) U-shaped image. (b) Probability density estimate of the edge map of the U-shape.



(a) (b) (c)

Fig. 2. Example of the edge probability density estimate for the corresponding continuous edge map. (a) Original image. (b) Continuous edge map obtained by $E(\mathbf{s}) = \nabla_x I(\mathbf{s})^2 + \nabla_y I(\mathbf{s})^2$. (c) Probability density estimate of the continuous edge map around the object boundary.

Explicitly, these choices are

$$E_{\text{cont}}(\mathbf{s}) = \begin{cases} (i): & E_{\text{binary}}(\mathbf{s}) * G(\mathbf{s}) \\ (ii): & \|\nabla I(\mathbf{s})\|^2 \\ (iii): & (ii) * G(\mathbf{s}). \end{cases} \tag{4}$$

For continuous edge maps, the density can be expressed as a variable-weight variable-bandwidth KDE as

$$p_{\text{edge}}(\mathbf{s}) = \sum_{i=1}^{N} w_i K_{\Sigma_i}(\mathbf{s} - \mathbf{s}_i) \tag{5}$$

where the weights w_i are obtained for each pixel as its normalized edge map value

$$w_i = \frac{E(\mathbf{s}_i)}{\sum_{j=1}^{N} E(\mathbf{s}_j)}. \tag{6}$$

Note that the binary edge map density is a special case of the latter, obtained by thresholding the weights appropriately to obtain binary weights for each pixel. In Fig. 2, an illustration of the edge density is presented obtained by using a continuous edge map for the aircraft image obtained from the *Berkeley Segmentation Dataset and Benchmark* [25].

3) Noise Robustness and Variable Bandwidth KDE: Robustness of the snake to outlier edge pixels, referred to as *fake edges*, in the image that might arise due to noise or texture is a desirable property. Such fake edges may adversely effect the progression of the snake as well as the quality of the final solution obtained by one. In the KDE framework, the influence of such isolated outlier data points can be suppressed by employing variable-bandwidth kernel estimates. Convenient methods for selecting data-oriented kernel widths typically involve K-nearest neighbor type approaches. For statistical consistency, K is usu-

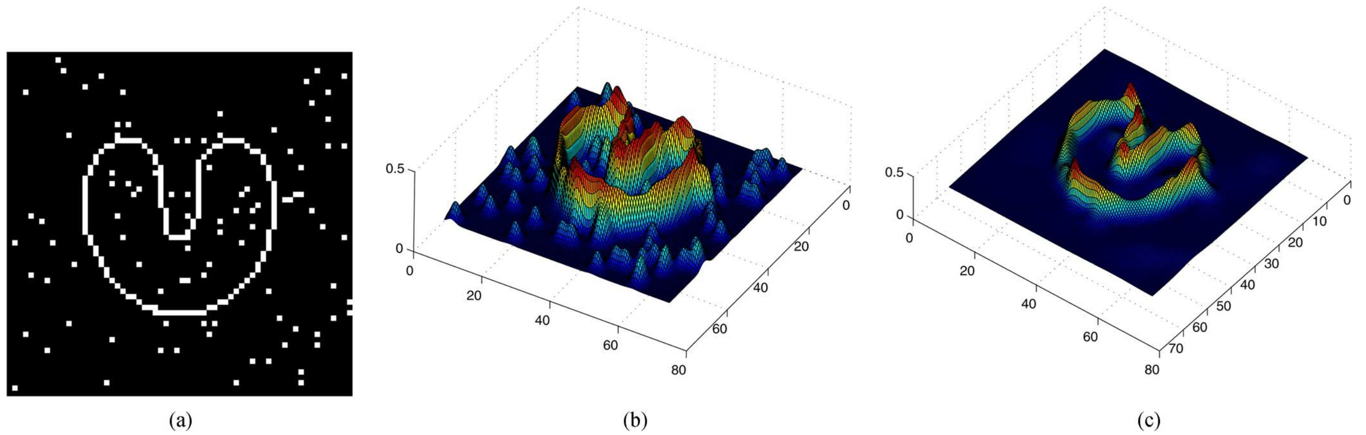


Fig. 3. Comparison of variable and fixed-size KDE for a noise edge map. (a) *Noisy* U-shaped image (b) Probability density of the noisy edge map with fixed-size KDE. (c) Probability density of the noisy edge map with variable size KDE.

ally selected to be a sublinear monotonically increasing function of N, the number of samples (pixels), such as $\sqrt{N}$.[1] A practical solution we have utilized in various domains successfully is to select the covariance of the kernel for sample i to be proportional to the KNN-covariance of its nearest neighbors, $\Sigma_i = \sigma^2 \mathbf{C}_i^{KNN}$, where σ^2 is optimized using leave-one-out maximum likelihood cross validation [21].

An illustration of how variable-bandwidth KDE eliminates spurious edge pixels and helps boundary identification is presented in Fig. 3. A noisy binary edge map is obtained by adding noise to a clean U-shaped image such that $\mathbf{s}_{\text{noisy}} = \mathbf{s} \oplus \mathbf{n}$, where $\oplus$ represents the logical or operator and the probability density of the noise $\mathbf{n}$ is

$$p_{\mathbf{n}} = 0.9\delta(\mathbf{n}) + 0.1\delta(\mathbf{n} - 1). \tag{7}$$

Edge density estimates using fixed-isotropic KDE and variable-anisotropic KDE are shown in the figure. Clearly, the latter provides a better cost function surface by smoothing out the density for the outlier edges and preserving the object boundary unchanged for detection purposes.

In the earlier approaches, noise robustness is generally obtained by convolving the original edge map with a Gaussian filter. However, while this low-pass filtering operation reduces noise, it also leads to a loss in the edge information by blurring weak edge regions. Choosing a variable kernel width individually for each sample, effectively corresponds to controlling the amount of the blurring adaptively throughout the image (in a spatially nonstationary manner). This provides noise reduction in the edge density estimate without introducing locally significant blur to the edges. On the negative side, variable-bandwidth KDE imposes a higher computational cost than its fixed-bandwidth counterpart.

4) Ridges of the Edge KDE Give the Object Boundaries: As it is clear from the illustrations above, the sought boundaries of the objects are identified by the ridges of the edge KDE and one simply needs to trace these ridges of interest. Pixel-level intensity-ridge tracking has been previously employed in medical image segmentation [19], [20]. In this approach, the image is mapped into a 3-D space consisting of the pixel coordinates, x, y, and the corresponding intensity $I(x, y)$. The segments of the image are considered to be relatively constant, and segmentation is achieved by tracing the ridges of the image intensity map using exhaustive search at pixel resolution. In contrast, the KDE-smoothing employed in this paper allows subpixel level ridge-tracing.

It can be shown that for the points on the ridges of the edge KDE: *i)* the local gradient of the edge KDE is parallel to an eigenvector of the local Hessian of the edge KDE evaluated at the same point, *ii)* the other eigenvector of the Hessian has a negative eigenvalue, that is, the ridge is a local maximum in the subspace orthogonal to itself [22]. Consequently, if one identifies any point on the ridge (such as a local maximum), the eigenvectors of the local Hessian can be traced using a numerical integration algorithm such as Runge–Kutta-4 [23], in order to identify the true optimal boundary of the objects. However, our experiments using this approach (not shown here) demonstrated that this approach is numerically unattractive due to high computational load and error accumulation at high curvature ridge locations.

B. Optimization Criterion

Given the active contour samples $\{\mathbf{s}_j^{\text{snake}}\}_{j=1}^{N_{\text{snake}}}$ and the density of the edge map $p_{\text{edge}}(\mathbf{s})$, our aim is to find a contour that *captures the structure* of the edge field $E(\mathbf{s})$ around the object to be segmented. We formulate this idea as maximizing the inner product between the probability density function of the snake $p_{\text{snake}}(\mathbf{s})$ and the probability density of the edge map $p_{\text{edge}}(\mathbf{s})$

$$\max_{\{\mathbf{s}^{\text{snake}}\}} J(\{\mathbf{s}^{\text{snake}}\}) = \max \int p_{\text{edge}}(\mathbf{s})\, p_{\text{snake}}(\mathbf{s})\, d\mathbf{s} \tag{8}$$

where the probability density of the snake $p_{\text{snake}}(\mathbf{s})$ is also evaluated as a KDE, using the samples of the snake

$$p_{\text{snake}}(\mathbf{s}) = \frac{1}{N_{\text{snake}}} \sum_{j=1}^{N_{\text{snake}}} K_{\Gamma_j}(\mathbf{s} - \mathbf{s}_j^{\text{snake}}) \tag{9}$$

[1] An asymptotically unbiased and consistent density estimate should satisfy $\lim_{N\to\infty} K = \infty$, $\lim_{N\to\infty} K/N = 0$

TABLE I
SUMMARY OF THE HIERARCHICAL FIXED-POINT ALGORITHM.

- Generate the edge field, $E(\mathbf{s})$.
- Select the kernel size and estimate the probability density of the edge field $p_{edge}(\mathbf{s})$ using (2) or (5).
- Select the initialization of the snake and estimate the probability density of the snake $p_{snake}(\mathbf{s})$ using (9)
- Use the iteration scheme given in (12) to iterate the points on the snake.
- Select a neighborhood threshold $th_{neighbor}$, which will define the resolution level of the snake - default value should be unity.
- For the points on the snake, build a graph of pairwise distances and find the points that have no neighbors for the given threshold.
- Select a perturbation step size μ and perturb these points towards M randomly selected directions. Typical values are $\mu = th_{neighbor}/5$ and $M = 5$. Optionally, if it's known that the interior or the exterior of the boundary provides a smoother cost function, this M randomly selected directions could be selected accordingly. This can be implemented by utilizing the inner product of the selected directions and the vector that connects the particular point to be perturbed to its location in the previous iteration.
- Use (12) to map these perturbed points to their projection on the contour. After convergence, add these points to the snake and include them in the neighborhood threshold calculation.
- Repeat until the predefined neighborhood threshold is satisfied at all points.

where N_{snake} is the number of points on the snake. Substituting (9) and (5) into (8), the plug-in KDE-based estimate of the objective function is obtained

$$J(\mathbf{s}^{\text{snake}}) = \sum_{j=1}^{N_{\text{snake}}} \sum_{i=1}^{N} \frac{w_i}{N_{\text{snake}}} K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{\text{snake}}). \qquad (10)$$

Note that this cost function is additive in terms of the samples of the snake. Therefore, near the optimal point along the ridge of the edge density, higher and lower sampling rates of the snake would lead to an accordingly denser or sparser evaluation of the optimality criterion along the ridge. The KDE of the edge map contains all smoothness information needed by the snake, which can be simply extracted by sampling the snake at a higher rate (as will be demonstrated later in the paper). In practice, utilizing an isotropic fixed-bandwidth for $p_{\text{snake}}(\mathbf{s})$ is sufficient (i.e., $\Gamma_j = \gamma^2\mathbf{I}$).

C. Fixed-Point Algorithm

In order to achieve faster convergence than gradient-like step-based algorithms, a fixed-point approach will be preferred in optimization algorithm design. To derive a fixed-point iteration for the samples of the snake, we equate the gradient of the optimization criterion by using the fact that for any fixed point of the density inner product cost function, the gradient of the inner product with respect to $\mathbf{s}^{\text{snake}}$ should be equal to zero. This yields the following:

$$\begin{aligned}\frac{\partial J(\mathbf{s})}{\partial \mathbf{s}_j^{\text{snake}}} &= \sum_{i=1}^{N} \frac{w_i(\Sigma_i+\Gamma_j)^{-1}}{N^{\text{snake}}} (\mathbf{s}_j^{\text{snake}} - \mathbf{s}_i) K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{\text{snake}}) \\ &= \mathbf{0}. \qquad (11)\end{aligned}$$

Reorganizing the terms and solving for $\mathbf{s}_j^{\text{snake}}$, the fixed-point update rule can be written as[2]

$$\mathbf{s}_j^{\text{snake}} \leftarrow \frac{\sum_{i=1}^{N} \mathbf{s}_i \, w_i(\Sigma_i + \Gamma_j)^{-1} \, K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{\text{snake}})}{\sum_{i=1}^{N} w_i(\Sigma_i + \Gamma_j)^{-1} \, K_{\Sigma_i+\Gamma_j}(\mathbf{s}_i - \mathbf{s}_j^{\text{snake}})}. \qquad (12)$$

This iteration is similar to that of the mean-shift and has been shown to be an EM-update [24], [18], thus, is convergent. Since this iteration propagates the snake samples to local maxima along the ridge, careful manipulation of the kernel size for the edge KDE will facilitate obtaining a sufficiently dense snake-sample distribution along the ridge. Specifically, one could utilize wide kernels to eliminate spurious maxima to facilitate locating the object and then switch to narrow kernels to generate more local maxima along the ridge to facilitate the interpolation of the boundary. Although, in our experiments, a need to implement this procedure has not been necessary, there might arise certain situations that would benefit from it. In practice, the iteration above converges at a rate proportional to the cube of the eigenvalue of the local Hessian in the vicinity of a local maximum. Consequently, along a relatively level ridge, one eigenvalue is significantly closer to zero (corresponding to the eigenvector pointing along the ridge) than the other. Due to the elimination of spurious edge maxima, the nonparametric snake does not suffer from poor capture range.

The fixed-point iterations above introduce a drawback: depending on the initial conditions, the original set of snake samples might be unable to progress into boundary concavities. The GVF definition overcomes this problem, and given suitable parameters, the GVF snake is able to progress into concavities. For the nonparametric snake, since the optimal interpolation idea of tracing the ridge by following the eigenvectors of the local Hessian is found to be numerically unattractive, an alternative solution to densely populate the snake is proposed next.

D. Fast and Robust Approach to Populate the Snake

The fixed-point iterations and the optimization criterion presented above can be utilized to increase the number of samples in the snake in order to densely populate the boundary once initial convergence is achieved by the original snake samples. The idea is to initialize multiple snake samples for each original sample around the corresponding convergence points and have the new samples converge to the boundary utilizing the same fixed-point iterations. The process can be repeated until a suitable criterion is satisfied (such as the maximum separation between two snake samples is less than T pixels). A step by step summary of the overall algorithm is given in Table I.

Once the stopping criterion is achieved, smooth interpolation of the final samples using various schemes is trivial to obtain a closed contour. The most commonly used interpolation methods in the literature include nearest neighbor interpolation, bilinear interpolation, bicubic interpolation, and spline interpo-

[2] In (12), the division denotes left multiplication with the inverse of the matrix given in the denominator. Specifically, for a spherical Gaussian, the matrix in the denominator will become a scalar.

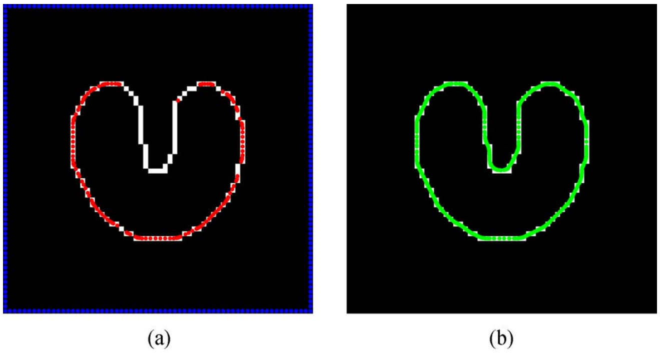


Fig. 4. Example of binary edge map results. (Left) Result of the fixed-point iterations and (right) the final result are shown. (a) Result of the fixed-point algorithm for the U-shaped image. (b) Final result after interpolations along the ridge of the pdf (see in color).

lation, which have found wide spread use in active contour field. For example, Xu and Prince mention that they use bilinear interpolation to obtain the active contour in a subpixel accuracy [8].

The underlying assumption behind utilizing interpolation of snake samples to obtain a closed contour is that there is a single object whose boundary is sought and this object occupies a connected region in the image. If the assumption of a single object fails, then the snake samples may identify boundaries of multiple objects and the closed contour obtained via interpolation will yield a wrong segmentation result. Since the proposed approach constructs the snake in a nonparametric manner one sample at a time, standard clustering algorithms (such as ones based on pairwise similarity) with appropriate model order selection mechanisms can be employed on the final snake samples to identify the number of distinct objects whose boundaries are identified. Therefore, the interpolation of the boundaries of these multiple objects can be carried out properly as usual. Since model order selection in clustering is an evolving research field, we do not investigate this proposition here in detail.

III. Experimental Results and Comparisons

A. Binary Edge Map—The U-Shape

We use the U-shaped image to show how capture range and boundary concavity issues are addressed by the proposed algorithm. This also provides the opportunity to compare our results with those obtained by the GVF snake. We initialize the nonparametric snake to the image boundary. It takes a few iterations for the fixed-point approach to find the boundary coarsely. Fig. 4 shows results obtained for this image, where the underlying density estimate of the binary edge field is previously presented in Fig. 1. Fig. 4(a) shows the convergence of the original snake samples after two iterations and Fig. 4(b) shows the final result after repeated runs of the fixed-point iterations as described in Table I.

B. Continuous Edge Map – Berkeley Segmentation Dataset

We present the performance of the proposed approach using a continuous edge map. The aircraft image used in this example is chosen from Berkeley Image Segmentation Benchmark [25]. The snake is initialized to the image boundary. Fig. 5 shows results obtained with the nonparametric snake for the continuous edge map. The corresponding continuous edge map and the estimate of the underlying probability density were presented in Fig. 2. Fig. 5(a) shows the initialization and the convergence of the original snake samples after a few fixed-point iterations. Fig. 5(b) shows the final result of the algorithm in Table I.

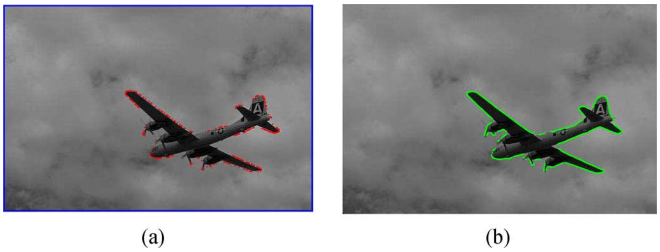


Fig. 5. Example of continuous edge map results. (Left) Result of the fixed-point iterations and (right) the final result are shown. (a) Result of the fixed-point algorithm for the BISB image. (b) Final result after interpolations along the ridge of the pdf (see in color).

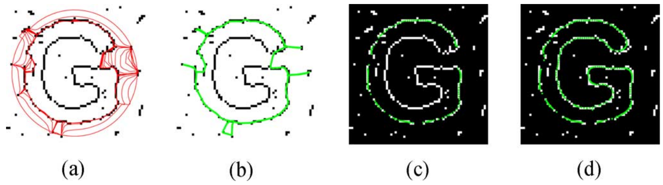


Fig. 6. Comparison of GVF snake and nonparametric snake on a noisy example with missing edges. (a) GVF snake iterations for the G-shaped image. (b) Final result for the GVF snake. (c) Result of the second iteration for the nonparametric snake. (d) Final result of the nonparametric snake using the hierarchical fixed-point algorithm (see in color).

C. Comparison With the GVF Snake—Noisy G-Shape

For the examples presented above, one can argue that GVF snake would obtain similar results with properly chosen parameters. This argument is generally true, if the *suitable* values of these parameter are known or can be derived from the image in the preprocessing stage. Still, these parameters control the shape priors of the snake globally, whereas the nonparametric snake is learning the shape parameters from the image, which inherently defines varying shape parameters throughout the image. In this G-shaped example, a 64×64 pixel-square G-shaped image corrupted with noise and missing edges is utilized to compare the performances of the nonparametric snake and the GVF snake. Results are presented in Fig. 6.

The main adverse implication of defining global smoothness parameters for the snake become most apparent in this example where a major concavity requires local high curvatures while a missing edge at a smooth portion of the boundary causes the globally highly curve snake to penetrate inwards due to an attraction by a spurious edge. The nonparametric snake does not suffer from this issue due to the possibility of selecting variable-bandwidth KDE for edge distributions, thus achieving better noise robustness, providing smooth connections along missing edge portions while allowing the propagation of the snake into concavities.

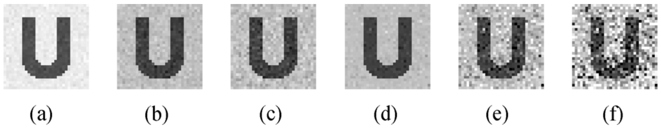


Fig. 7. U-shape buried in different types and levels of noise. (a) Gaussian noise $p_{\mathrm{noise}} = N(0,5)$. (b) Gaussian noise $p_{\mathrm{noise}} = N(0,10)$. (c) Gaussian noise $p_{\mathrm{noise}} = N(0,15)$. (d) Laplace noise $p_{\mathrm{noise}} = 5L(0,1)$. (e) Laplace noise $p_{\mathrm{noise}} = 10L(0,1)$. (f) Laplace noise $p_{\mathrm{noise}} = 15L(0,1)$.

TABLE II

	P	R	Run time
$p_n = N(0,5)$	0.95 ±0.02	0.96 ±0.02	2.875 s.
$p_n = N(0,10)$	0.93 ±0.03	0.92 ±0.04	3.954 s.
$p_n = N(0,15)$	0.86 ±0.03	0.83 ±0.04	4.054 s.
$p_n = 5L(0,1)$	0.91 ±0.03	0.90 ±0.04	2.921 s.
$p_n = 10L(0,1)$	0.81 ±0.04	0.76 ±0.06	3.893 s.
$p_n = 15L(0,1)$	0.69 ±0.08	0.64 ±0.09	5.436 s.

Running time results are for a pentium4 2.8 GHz processor

Specifically, for this example, we used anisotropic Gaussian kernels whose covariances for each edge pixel is given by $\Sigma_i = \sigma^2 C_i^{KNN}$, where C_i^{KNN} is the covariance of K nearest neighbor points in the binary edge map, and σ^2 is a global scaling constant, optimized using leave-one-out maximum likelihood criterion. The hierarchical fixed-point algorithm of Table I is used with parameters $\mathrm{M} = 5$ and $\mu = 0.2$).

D. Continuous Edge Maps With Different Noise Distributions

In this section, we present the effect of a noisy continuous edge map on the performance of the nonparametric snake. Results for Gaussian and Laplace distributed intensity noise for three different SNR levels are shown. A variable-bandwidth KDE with anisotropic Gaussian kernel functions as described above are utilized. Fig. 7 presents sample images for the U-shape buried in noise for different noise types and levels. Table II provides a quantitative analysis of performance utilizing the precision (P) and recall (R) values for each noise type and SNR, averaged over ten Monte Carlo runs, as well as the run-time of the complete algorithm until convergence on a typical desktop computer. If there is no major concavity in the boundary, there is no significant difference in running time. However, for this particular example, missing edges make it more difficult to progress into concavities and overall running time increases due to more reinitializations as described in Table I.

IV. Conclusion

In the original snake formulation, the required parameters can only be set empirically, that is, the algorithm is run a number of times until a satisfactory performance is reached. This drawback can be overcome by defining both energy functions in the same domain, and we propose a nonparametric approach that exploits the underlying edge probability density. In this paper, we present a nonparametric approach to the active contour problem. We design the algorithm such that it first exploits the edge field without considering the shape. This enables the proposed nonparametric snake to define local shape priors. The formulation based on kernel density estimation presents a convenient parameter selection framework and outlier robustness.

Our future work will include a formulation that is similar to snakes in practice, but allows multiple segments controlled by an adjustable granularity level. For images containing multiple complex boundaries, initializing the nonparametric snake samples to every pixel of the image and clustering the points that belong to the same boundary after convergence might prove to be more efficient than the re-initialization procedure proposed in this paper.

Acknowledgment

The authors would like to thank J. C. Principe, S. Rao, M. E. Sargin, and E. Tola for valuable discussions.

References

[1] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[2] K. S. Fu and J. K. Mei, "A survey on image segmentation," *Pattern Recognit.*, vol. 13, pp. 3–16, 1981.

[3] R. Pal and S. K. Pal, "A review in image segmentation techniques," *Pattern Recognit.*, vol. 26, pp. 1277–1294, 1993.

[4] D. Comaniciu and P. Meer, "Mean shift: A robust approach towards feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, pp. 321–331, 1987.

[7] F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 6, pp. 617–634, Jun. 1993.

[8] C. Xu and P. L. Prince, "Snakes, shapes and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998.

[9] R. Ronfard, "Region-based strategies for active contour models," *Int. J. Comput. Vis.*, vol. 13, pp. 229–251, 1994.

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

[11] B. Leroy, I. Herlin, and L. D. Cohen, "Multiresolution algorithms for active contour models," in *Proc. 12th Int. Conf. Anal. Optim. Syst.*, 1996, pp. 58–65.

[12] L. D. Cohen and I. Cohen, "Finite element methods for active contour models and baloons for 2D and 3D images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pt. 11, pp. 1137–1147, Nov. 1993.

[13] C. Davatzikos and J. L. Prince, "An active contour model for mapping the cortex," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 65–80, Jan. 1995.

[14] A. J. Abrantes and J. S. Marques, "A class of constrained clustering algorithms for object boundary extraction," *IEEE Trans. Image Process.*, vol. 5, no. 11, pp. 1507–1521, Nov. 1996.

[15] E. Parzen, "On the estimation of a probability density function and the mode," *Ann. Math. Statist.*, vol. 32, pp. 1065–1076, 1962.

[16] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

[17] H. Weinert, Ed., *Reproducing Kernel Hilbert Spaces: Applications in Statistical Signal Processing* Stroudsburg, PA: Hutchinson Ross, 1982.

[18] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[19] J. C. Crowley and A. C. Parker, "A representation for a shape based on peaks and ridges in the difference of low-pass transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 2, pp. 156–170, Feb. 1984.

[20] L. M. Lifshitz, "A multiresolution hierarchical approach to image segmentation based on intensity extrema," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 6, pp. 529–540, Jun. 1990.

[21] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[22] D. Erdogmus and U. Ozertem, "Locally defined principal surfaces," presented at the ICASSP, 2007.

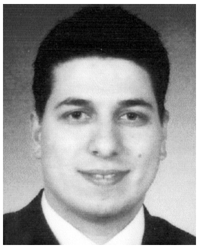
[23] J. D. Lambert and D. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. New York: Wiley, 1991, ch. 5.

[24] M. A. Carreira-Perpinan, "Gaussian mean shift is an EM algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[25] D. Martin, C. Fowlkes, D/ Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int. Conf. Computer Vision*, 2001, vol. 2, pp. 416–423.

Umut Ozertem (S'06) received his B.S. degree in electrical and electronics engineering in 2003 from the Middle East Technical University, Turkey.

After working at Tubitak-BILTEN between August 2003 and July 2004 under the supervision of A. Alatan, he joined the Adaptive Systems Lab, CSEE Department, Oregon Health and Science University, Portland, as a Ph.D. student. He is currently working with D. Erdogmus, and his research interests include statistical signal processing and machine learning with a focus on information theoretic methods.



Deniz Erdogmus (M'00) received the B.S. degree in electrical and electronics engineering and the B.S. degree in mathematics, both in 1997, and the M.S. degree in electrical and electronics engineering in 1999 from the Middle East Technical University, Turkey, and the Ph.D. degree in electrical and computer engineering from the University of Florida (UF), Gainesville, in 2002.

He was a Research Engineer at TUBITAK-SAGE, Turkey, from 1997 to 1999, and he was a Research Assistant and a Postdoctoral Research Associate at UF from 1999 to 2004. Currently, he is an Assistant Professor at the CSEE and BME Departments of the Oregon Health and Science University, Portland. His research focuses on information theoretic and nonparametric techniques in statistical signal processing and thier applications to biomedical problems.

Dr. Erdogmus has served on the editorial and scientific boards of a number of journals and conferences. He was the recipient of the IEEE-SPS 2003 Best Young Author Paper Award and 2004 INNS Young Investigator Award.