# A Nonparametric Approach for Active Contours

Umut Ozertem, Deniz Erdogmus

***Abstract*— Active contours are commonly used in many image segmentation applications. There are different active contour definitions, but all active contour definitions in the literature use parametric forms to determine the shape priors or adjust the weighting of internal and external forces acting on the active contour. However, the evaluation or estimation of the *optimal* values of these parameters is impossible in a general sense, and the algorithms are run with different parameters until a satisfactory result is obtained. To get rid of this exhaustive parameter search, we approach the same problem in a nonparametric way to translate the problem of seeking good values of these unknown parameters into seeking for a good density estimate. We tested the proposed method and compared with earlier approaches and obtained better results.**

## I. Introduction

Image segmentation is a fundamental problem in image processing with applications in object recognition, image and video coding, filtering of noisy images, to name a few. The image segmentation problem is to partition the image into distinct homogenous regions, which is an ill-posed definition that varies for depending on the application. The homogeneity in the definition does not strictly require homogeneity in the intensity space and can be defined in any feature space. The earliest methods in the image segmentation literature are based on edge detection[1]. Edge-based methods are computationally efficient and they use directional derivatives to detect the edges of the image, and they combine these edges to obtain the object contours. The most obvious drawback of edge-based approaches is that they need the intensity values themselves to be a useful feature for segmentation and results obtained with these methods are very sensitive to the predetermined parameter values. The methods that overcome this parameter selection problem are usually some deviants of clustering algorithms. These are known as clustering-based segmentation approaches, which bring the generalization properties of data clustering into the field of image segmentation [2], [3].

A relatively newer approach is active contours, so-called snakes [4], [5], [6]. Active contours define a way of combining edges in the image using the predefined shape priors. The total force acting on a active contour is a blend of the internal and external forces, where internal forces depend on the shape of the contour itself, and external forces are evaluated using the edges of the image. There are two main problems in the active contour literature: narrow initialization range, and progressing into boundary concavities. The low capture range problem is that the initialization of the active contour has to be in a narrow neighborhood of the object boundary. In some applications, snakes are initialized to the output of another segmentation algorithm to solve this problem. Techniques that increase the capture range includes multiresolution and distance potentials [8]. The second well-known problem is that the snakes have difficulties in progressing into concavities along the boundary. There are many approaches to solve this problem, including directional attractions [10], control points [9], pressure forces [7]. However, the most satisfactory results are obtained with the *gradient vector flow* (GVF) formulation by Xu and Prince [6], which also solves the capture range issue very effectively. GVF formulation provides a wide capture range and an ability to progress into boundary concavities; however, selection of required parameters is an unsolved problem. As with most parametric methods, the usual way of seeking the optimal result is to run the algorithm several times for a set of different parameter values until a satisfactory performance is obtained.

In this paper, we define the problem in a nonparametric way that translates the parameter search problem into a density estimation problem. Density estimation is a very well-researched field, and the connection that we present gives way to utilization of many well-known density estimation techniques in the active contour field. In this paper, we mainly focus on kernel density estimation to derive an algorithm, but this selection can easily be relaxed and other density estimation methods can also be utilized for the same purpose.

## II. Kernel Density Estimation

In many applications, determining a suitable parametric family is a tedious task. Data probability density functions may take complex forms, which makes the parametric density estimation methods impractical. Nonparametric density estimation methods offer a number of techniques that overcomes this issue. For example, estimators based on sample spacing are not continuously differentiable, and they are not suitable for gradient based adaptation. However, for a continuously differentiable kernel function, estimators based on KDE offer a continuously differentiable density estimate. The most important issue in KDE is to determine a suitable kernel function, and there is a wide literature about how to approach this problem [11], [12]. The method to select the kernel depends on the application, and the characteristics of the input data and the feature space. We present our results with commonly used kernel function selections, since the aim of this paper is to introduce the concept, not to build a complete system for a specific application. More specifically, variable

Umut Ozertem and Deniz Erdogmus are with the Department of Computer Science and Electrical Engineering, Oregon Health and Science University, Portland, OR, USA email: {ozertemu,deniz}@csee.ogi.edu).This work is partially supported by the NSF grants ECS-0524835, and ECS-0622239.

kernel density estimation finds natural connections to the well-known problems in the active contour field.

## III. The Nonparametric Snake

In this section, we develop the nonparametric snake. We start with the selection of the feature space, and the definition of the optimization objective. Then we propose a fixed point iteration scheme to develop the algorithm.

Consider an image $I(x, y)$. To define the objective criterion we map the image into the feature space $\mathbf{s}$, where every vector in this space collects the *x*, and *y* pixel coordinates in the image.

$$\mathbf{s} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

Snakes use the edge map of the image, and the edge map can be defined either in a continuous or a binary form. A binary edge map $E(\mathbf{s})$ is given as

$$E(\mathbf{s}_{binary}) = \begin{cases} E(\mathbf{s}_i) = 1 & : \ \mathbf{s}_i \ is\ an\ edge\ pixel \\ E(\mathbf{s}_i) = 0 & : \ otherwise \end{cases} \tag{2}$$

whereas popular continuous forms in the literature include

$$E_{cont}(\mathbf{s}) = \begin{cases} (i) & : \ E_{binary}(\mathbf{s}) * G(\mathbf{s}) \\ (ii) & : \ \|\nabla_x I(\mathbf{s})^2\| + \|\nabla_y I(\mathbf{s})^2\| \\ (iii) & : \ (ii) * G(\mathbf{s}) \end{cases} \tag{3}$$

where $G(\mathbf{s})$ is a smoothing function, which is usually selected as a Gaussian [6]. Our aim is to estimate the probability density of the edge map, which can be achieved by

$$p_{edge}(\mathbf{s}) = \sum_{i=1}^{N} w_i K_{\Sigma_i}(\mathbf{s} - \mathbf{s}_i) \tag{4}$$

where $N$ is the number of points in the image, $K_{\Sigma_i}(\cdot)$ is a Gaussian kernel with covariance $\Sigma_i$, and the weights $w_i$ are given as the edge map value of the corresponding point divided by the sum of the edge map values over all the data points.

$$w_i = \frac{E(\mathbf{s}_i)}{\sum_{i=1}^{N} E(\mathbf{s}_i)} \tag{5}$$

Note that the density estimate given here is for general form of continuous edge map, where in the special case of binary edge map the weights are binarized. For generality, we leave the kernel function with the subscript *i* for variable size KDE.

Given the samples of the snake $\{\mathbf{s}_j^{snake}\}_{j=1}^{N_{snake}}$, and the density of the edge map $p_{edge}(\mathbf{s})$, the objective is the capture the structure of the $E_{edge}(\mathbf{s})$. We formulate this idea as maximizing the inner product between the probability density function of the snake $p_{snake}(\mathbf{s})$, and probability density of the edge map $p_{edge}(\mathbf{s})$.

$$\max_{\{\mathbf{s}^{snake}\}} J(\{\mathbf{s}^{snake}\}) = \max \int p_{edge}(\mathbf{s})\ p_{snake}(\mathbf{s})\ d\mathbf{s} \tag{6}$$

The probability density of the snake can be estimated in a similar way using KDE as,

$$p_{snake}(\mathbf{s}) = \frac{1}{N_{snake}} \sum_{j=1}^{N_{snake}} K_{\Lambda_j}(\mathbf{s} - \mathbf{s}_j^{snake}) \tag{7}$$

where $N_{snake}$ is the number of points on the snake. To obtain a sample estimate of the cost function, one can substitute (4), and (7) into (6), which yields

$$J(\mathbf{s}^{snake}) = \sum_{j=1}^{N_{snake}} \sum_{i=1}^{N} \frac{w_i}{N_{snake}} K_{\Sigma_i + \Lambda_j}(\mathbf{s}_i - \mathbf{s}_j^{snake}) \tag{8}$$

This cost function is solely defined as a function of the edge map of the image, and does not contain any shape information. The shape information will be exploited by optimizing this function in an iterative manner in the neighborhood of the convergence, which we will introduce after presenting the optimization scheme.

To find the optimizer of (8), we propose a fixed-point algorithm. Alternatively, gradient ascent based approaches can also be utilized here, but fixed-point methods have a relatively fast convergence rate. Convergence of a fixed-point algorithm can be proven using Hillam's Theorem for any Lipschitz continuous function as the cost function we have in our application.

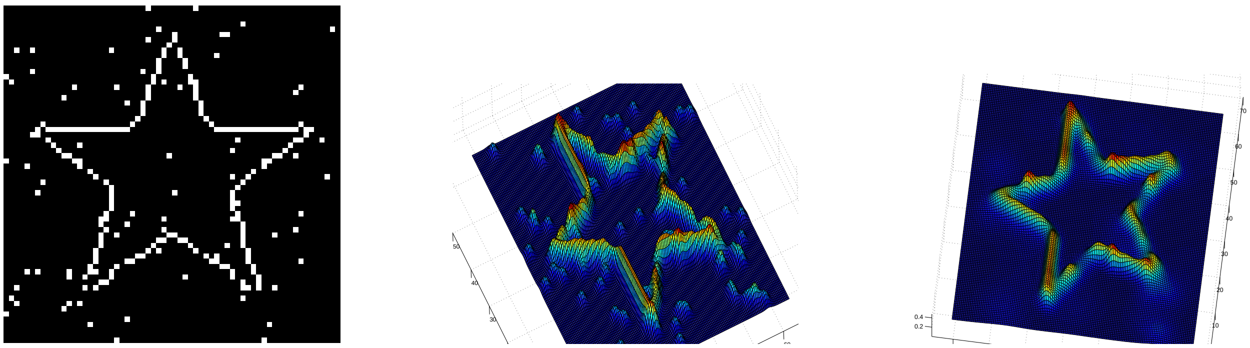
To derive the fixed-point algorithm one should use the fact that for any fixed-point of the objective function, the gradient with respect to $\mathbf{s}^{snake}$ should be equal to zero. This yields

$$\frac{\partial J(\mathbf{s})}{\partial \mathbf{s}_j^{snake}} = \sum_{i=1}^{N} \frac{w_i (\Sigma_i + \Lambda_j)^{-1}}{N^{snake}} (\mathbf{s}_j^{snake} - \mathbf{s}_i) K_{\Sigma_i + \Lambda_j}(\mathbf{s}_i - \mathbf{s}_j^{snake}) \tag{9}$$

Reorganizing the terms and solving for $\mathbf{s}$, one can write the fixed point update as

$$\mathbf{s}_j^{snake} = \frac{\sum_{i=1}^{N} \mathbf{s}_i \ \ w_i (\Sigma_i + \Lambda_j)^{-1} \ \ K_{\Sigma_i + \Lambda_j}(\mathbf{s}_i - \mathbf{s}_j^{snake})}{\sum_{i=1}^{N} w_i (\Sigma_i + \Lambda_j)^{-1} \ \ K_{\Sigma_i + \Lambda_j}(\mathbf{s}_i - \mathbf{s}_j^{snake})} \tag{10}$$

Although the iterations converge fast, this optimization scheme has a major shortcoming. Depending on the selection of the kernel size, the fixed-point optimization has the risk of being unable to progress into boundary concavities, which is one of the most important problems in the active contour literature. Recursive runs of the algorithm with the proper initializations effectively solve this problem. Before we go into the details of the implied shape parameters and the iterated fixed-point algorithm, we will present a simple example for the density estimate of the edge field obtained using fixed and variable-size kernels. Noise robustness is one of the most important design issues in the active contour literature and variable size KDE methods find a natural solution to this problem. Convolving the original edge map with a smoothing function is the most common technique to increase noise robustness. In this approach, the robustness to fake edges increases with amount of introduced smoothness. However, this smoothing not only eliminates the *fake* edges, but also distorts the sought object boundary causing it to become smoother. In our nonparametric approach, this problem translates to the selection of kernel size. Widening the kernel size result in a better noise robustness level, but the detailed information of the boundary is lost. Utilizing a variable size KDE solves this problem automatically. In KDE literature, the aim of assigning variable kernel sizes to each data point is to increase the outlier robustness. The main idea



(a) (b) (c)

Fig. 1. An example of the edge probability density estimate obtained for fixed, and variable size.(a) The edge field of the star-shape image (b) pdf of the edge field using fixed size KDE (c) pdf of the edge field using variable size KDE

TABLE I

SUMMARY OF THE ITERATED FIXED-POINT NONPARAMETRIC SNAKE 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 (4).
- Select the initialization of the snake and estimate the probability density of the snake $p_{snake}(\mathbf{s})$ using (7)
- Use the iteration scheme given in (10) 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.
- Select a perturbation step size $\mu$ 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.
- Iterate these points using (10) 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.

is to select the kernel size for every data point in a way that it is proportional to the probability of that sample's being an outlier. This probability can be implemented in many ways. The most intuitive and simple choice is to multiply each kernel size with the average distance to *K* nearest neighbors. In this way, for the samples that do not have many close neighbors and are presumably fake edges, wider kernels will be used. Variable size KDE introduces a data dependent variable smoothing throughout the image, which helps to eliminate the outlier edges while keeping the object boundary almost unchanged. Figure 1a shows the boundary of a star-shape object for a noisy case, where we have outlier edges. Figure 1b, and Figure 1c show the probability density of the edge field for fixed and variable size kernel functions, respectively. For this example the kernel size is selected to be $\sigma^2 = 1$ for the fixed size KDE, and $\sigma_i^2 = d_i^{KNN}$ for the variable size KDE. $d_i^{KNN}$ is the average distance to K nearest data points, and for this particular example we selected this parameter as $K = 5$. One can see that the variable size KDE effectively removes the outliers from the edge probability density; hence, from the objective criterion without distorting the edges. In the next section we will introduce the characteristics of the underlying shape priors as we present the experimental results for our algorithm.

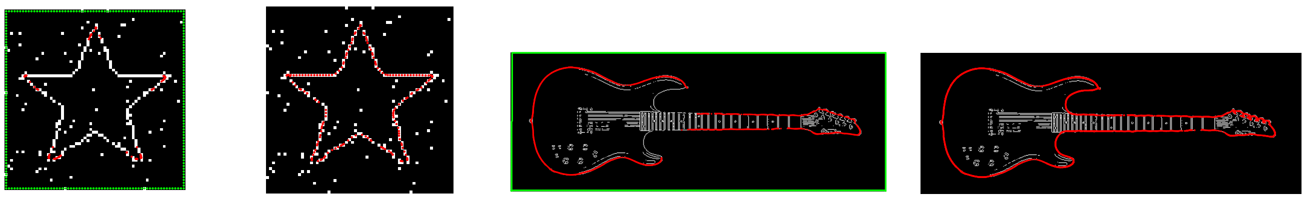
## IV. Experimental Results

In this section we will present results on the $64 \times 64$ star-shape image, and another more challenging real-data example. For both examples, we initialize the active contour to the boundary of the image, since the proposed nonparametric snake does not have any capture range problems at all. The iterations given in (10) converges to a local optimizer of the cost function in (6). At this point, note that the global optimization of this cost function would give nothing but the edge field itself, and is meaningless. Our aim is to find the local optimizer of the cost function in the neighborhood of a given initialization, which is what makes the initialization meaningful.

As one can see the experimental results, the proposed iteration scheme may introduce a problem on boundary concavities. To solve this issue, we perturb the points on the snake after convergence and use these points as an initialization until the all points form a connected pattern. The algorithm treats the missing edges and the concavities

(a) (b)

Fig. 2. The original guitar image (a), the corresponding edge field (b), and GVF iterations on the edge map are shown



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

Fig. 3. Initialization and the results for the first iteration of algorithm for star (a) and guitar (c) images, and the final results of the algorithm for star (b) and guitar images (d) are shown.

in exactly the same way, where the underlying edge density forces the active contour to progress into the concavity, or to get connected smoothly. This can be interpreted as defining shape priors in a local fashion. A summary of the algorithm is presented in Table 1.

In Figure 2, we utilize GVF snake to compare the results in a challenging example, where we distort the original edge map to obtain missing edges in a case we have boundary concavities. Since GVF defines the shape parameters globally, the parameters that would make the snake able to progress into the concavities make the points on the contour very sensitive to noise in the vicinity of missing edges. One can see this effect in Figure 2b. The parameters that make the active contour progresses into the big concavities prevents the missing edges from getting combined smoothly. Similarly, another set of parameters may give the active contour the ability to combine these edges in a smooth manner; however, this parameter set would presumably stop the progress of the snake into the concavities. Exploiting the underlying edge probability density, our nonparametric approach combines these missing edges smoothly, while having no problem in progressing into concavities. Figure 3 shows the intermediate results for the first initialization, and the final results after the iterated algorithm for the star-shape and the guitar images.

## V. Discussion

There are many different models for active contours in the literature, and all of these different definitions require properly selected parameters to achieve satisfactory results. In this paper, we exploit the underlying edge information and provide a nonparametric alternative. Due to the nature of the density estimation, the implied shape parameters are defined locally, which gives a more powerful tool as compared to the parametric approaches, where these parameters are defined globally. While providing this property, the proposed method also does not introduce any shortcoming on solving known problems in the active contour field. Issues like capture range, and progressing into boundary concavities are addressed by the proposed nonparametric approach.

## References


[1] J. F. Canny, "*A Computational Approach to Edge Detection*", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-698, 1986.

[2] D. Comaniciu, P. Meer, "*Mean Shift: A Robust Approach towards Feature Space Analysis*", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 603-619, 2002.

[3] J. Shi, J. Malik, "*Normalized Cuts and Image Segmentation*", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 888-905, 2000.

[4] M. Kass, A. Witkin, D. Terzopoulos, "*Snakes: Active Contour Models*", International Journal of Computer Vision, vol.1, pp. 321-331, 1987.

[5] F. Leymarie, M. D. Levine, "*Tracking Deformable Objects in the Plane using an Active Contour Model*", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 617-634, 1993.

[6] W. Xu and P. L. Prince, "*Snakes, Shapes and Gradient Vector Flow*", IEEE Transactions on Image Processing, 1998.

[7] L. D. Cohen, "*On Active Contour Models and Baloons*", CVGIP: Image Understand., vol. 53, pp. 211-218, 1991.

[8] B. Leroy, I. Herlin, L. D. Cohen, "*Multiresolution Algorithms for Active Contour Models*", $12^th$ Int. Conf. Analysis and Optimization of Systems, pp. 58-65, 1996.

[9] C. Davatzikos, J. L. Prince, "*An Active Contour Model for Mapping the Cortex*", IEEE Transactions on Medical Imaging, vol. 14, pp. 65-80, 1995.

[10] A. J. Abrantes, J. S. Marques, "*A Class of Constrained Clustering Algorithms for Object Boundary Extraction*", IEEE Transaction on Image Processing, vol. 5, pp. 1507-1521, 1996.

[11] E. Parzen, "*On the Estimation of a Probability Density Function and the Mode*", Annals of Mathematical Statistics, vol. 32, pp. 1065-1076, 1962.

[12] B. W. Silverman, "*Density Estimation for Statistics and Data Analysis*", Chapman and Hall, London, 1986.